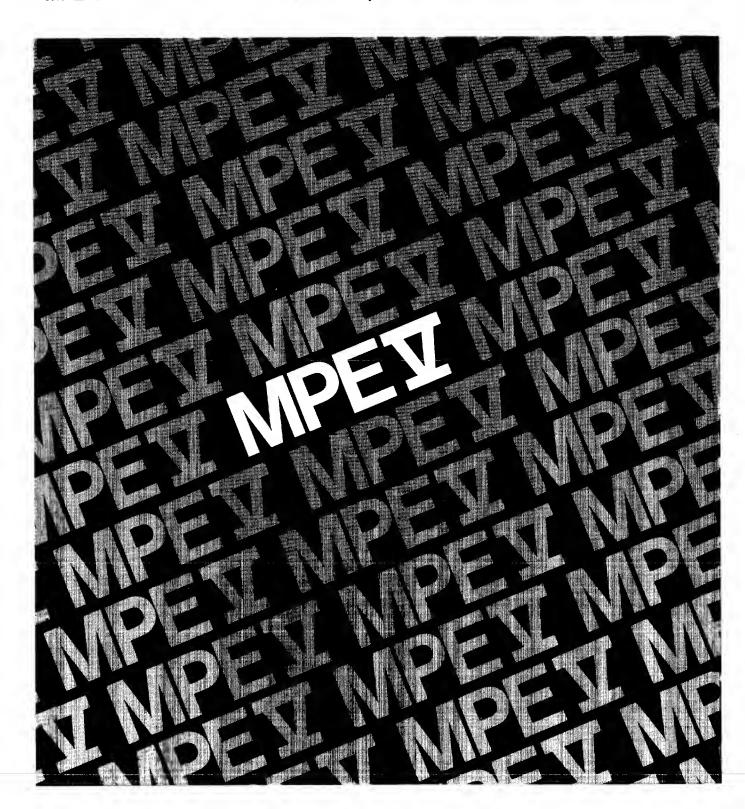
# **HP 3000 Computer Systems**



MPE V Tables Manual for MPE V/E, Version G.00.00



### **HP 3000 Computer Systems**

# MPE V TABLES MANUAL for MPE V/E, Version G.00.00



19447 PRUNERIDGE AVENUE, CUPERTINO, CA 95014

Part No. 32033-90010 E2412

Printed in U.S.A. 09/84

### CAUTION

The normal checks and limitations that apply to the standard MPE users are bypassed in Privileged Mode. It is possible for a Privileged Mode program to destroy file integrity including the MPE operating system software itself. Upon request Hewlett-Packard will investigate and attempt to resolve problems resulting from the use of Privileged Mode code. This service is available on a materials billing basis. However, time and Hewlett-Packard will not support, correct, or attend to any modifications of the MPE operating system software.

### NOTICE

The information contained in this manual is proprietary and may not be disclosed to third parties. The terms and conditions of release of this information to the recipient are governed by the agreement executed at and between Hewlett-Packard and the recipient regarding the HP 3000 internal MPE V System Table Structures, including the MPE V Tables Manual for MPE V Operating System.

Hewlett-Packard does not warrant the accuracy or completeness of the information contained in this manual. Hewlett-Packard may make revisions to the MPE operating system or CPU firmware that may affect the information and specifications contained herein. Hewlett-Packard shall have no obligation to provide the recipient with notification of revisions, corrections, or updates regarding the information contained herein.

Hewlett-Packard shall not be responsible for any loss or damage to the recipient, its customers, or any third party caused by the recipient's use of the information provided herein.

# LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the current edition, and lists the dates of all changed pages. Unchanged pages are listed as "ORIGINAL". Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars and dates remain. No information is incorporated into a reprinting unless it appears as a prior update.

First Edition	September 1984
Effective Pages	Date
ATT	SEP 1984

## PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated. No information is incorporated into a reprinting unless it appears as a prior update.

The software date code number printed alongside the date indicates the version level of the software product at the time the manual edition or update was issued. Many product updates and fixes do not require manual changes, and conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

First Edition . . . . . . . . SEP 1984 . . . . . . . . . . . . . . . G.00.00

# **CONTENTS**

PREFACE			 					xix
CHAPTER 1 MEMORY LAYOUT								
Fixed Low Memory (Series 44/48/64/68)			 					1-1
System Global Area								
SysGlob Extension								
SYSDB Words								
SysGlob Word Definitions								
Allow Mask Format								
Logging Related Locations								
FLAGX								
Process Stop List General Layout								
Entry Format								
Preassigned Entries								
Initial Memory Allocation			 					. 1-22
Bank 0								
Bank 1								
		•	 •	·		-		
CHAPTER 2 MEMORY MANAGEMENT TABLES								
G 4 T. 11. St								2 1
Segment Table Structure								
Pointers and DST #'s of Segment Table Components								
Standard Object Identifier Format								
DST Entry Formats								
CST Entry Formats			 					2-5
ST Entry Field Descriptions								
CSTBLK Format								
Program Blocks and the CSTXMAP								
CST Extension Block Entry Format								
Fixed DST Entry Assignments								
Swap Tables								
SWAPTAB Entry 0 Format								
SWAPTAB Unassigned Entry Format								
Segment Locality Lists (SLL)								
SLL Header Format			 . ,					. 2-14
SLL List Entry Format								
Special Request Table			 					. 2-16
Main Memory Region Headers and Trailers								
Global Region Trailer								
Global Region Header (Available Regions)								
Subregion Header (Available Regions)								
Global Region Header (Reserved Regions)								
Subregion Header (Reserved Regions)								
Subregion Header (Cached Regions)								
Region Header and Trailer Field Descriptions								
Space Allocation Structures	٠.	٠.	 	٠			 •	. 2-26
CHAPTER 3 DISC LAYOUT								
System Disc Layout								
Disc Label (Sector 0 of Disc)			 		, ,			3-4

System Volume	. 3-4
Serial Volume	. 3-6
Master Volume	
Slave Volume	
Defective Tracks Table (Sector 1 of Disc)	3-12
Defective Sector Table (DSCT Sector 1 of Disc)	3-13
Reserved Area Bit Map (Sector 4 of the System Disc)	3-14
Disc Cold Load Information Table (Sectors 28-30)	3-15
INITIAL Program CST Map	3-20
SYSDUMP/INITIAL Communication Record	3-21
Cold Load Information Table Extension	3-23
Virtual Disc Space Management Structures	3-24
Virtual Disc Space Management Table	3-24
General Structure	3-24
VDSMTAB Entry 0 Format	
VDSMTAB General Entry Format	3-26
Volume Table	3-27
Typical Private Volume Entry	
Typical System Volume Entry	2-20
2) production of the control of the	3-27
CHAPTER 4 DIRECTORY	
Introduction to the Directory	. 4-1
Overview of Directory	. 4-2
Directory Data Segment	
Directory Pointer Area [DA or DB] DST=20(10)	. 4-5
Directory Space Data Segment (DIRSDS)	4-6
Directory Structure	4-10
Directory Definitions	4-11
Index Block Prefix (10 Words)	4-11
Index Entry (6 Words)	4-12
Account Entry (%36 Words)	4-12
Group Entry (51% Words)	4-14
File Entry (File Pointer) (6 Words)	4-16
User Entry (19 Words)	4-17
User Attributes/Capabilities	4-17
Volume Set Definition Entry	4-10
GVSLINKAGE	
GVSINFO	
GVSVOLFLAGS	4-20
GVSVOLINFO	4-20
Volume Set Class Entry	
GVCLINKAGE	4-21
GVCINFO	4-22
Volume Mask Format	4-22
CHAPTER 5 LOCK RESOURCES	
SIR# Allocation DST %53	. 5-1
Sir's Ordered by Sir Number	5_1
Sir's Ordered by Ranking	5-2

SIR Table Information	
SIR Entry Formats	
RIN Table General Layout (Initialized State)	
Allocation and Locking of Local RINS	
Allocation and Locking of File RINS	
Allocation and Locking of Global RINS	. 5-8
CHAPTER 6 FILE SYSTEM	
	<i>(</i> 1
File System Overview	
Buffers	
Table Formats	
File System Section of PCBX (PXFILE)	
Overhead	
PXFILE Control Block Table (PXFCBT)	
Available Block	
Active File Table (AFT)	
File Control Block Table (CBTAB)	
Overhead	
Vector Table	
Control Block Area	
Access Control Block (ACB)	
Logical Access Control Block (LACB)	
Physical Access Control Block (PACB)	
File Control Block (FCB)	
File Label (FLAB)	
File Multi-Access Vector Table (FMAVT) DST(%54)	
Zero Entry Format	
Typical Entry Format	6-53
System Global Area (SYSGLOB)	
SIRs, Locks, and Deadlocks	6-54
Shared CBT DST	6-55
CHAPTER 7 PROCESS TABLES	
	<b>.</b>
Process Control Block Table Structure and Format	
Fixed Cells Related to PCB	
PCB Entry 0 Format	
Unassigned PCB Entry Format	
Assigned PCB Entry Format	
PCBX Structure and Format	
PCBX General Structure	
PXGLOB Format	
PXFIXED Assignments	
a saa	7-14
<del></del>	7-15
	7-16
	7-17
FORTRAN Logical Unit Table (FLUT)	7-18

### **CHAPTER 8 JOB TABLES** Job Master Table Structure (JMAT).....8-2 Job Master Table Entry (JMAT)......8-4 Job Directory Table (JDT).....8-14 Job Control Word Table (JJCW).....8-18 CHAPTER 9 RELOCATABLE OBJECT CODE Header Type 11..... 9-29

Entry Point Directory		9-32
Typical Directory Entry		9-33
Procedure Information Block		9-34
Headers		9-35
CHAPTER 10 PREPARED OBJECT CODE		
Program File Format		10-1
Flags		10-3
Flags2		
CST Remapping Array		
Segment Descriptor Array		10-4
Global Area Format		10-4
External List		10-5
Entry Point List		10-6
Code Segment With Patch Area		10-7
Patch Area		10-7
PMAP Information		10-8
PMAP Type Table		10-8
PMAP Records		10-9
Type 0 Segment PMAP Record		10-9
Type 1 Procedure PMAP Record		10-9
Type 2 Secondary Entry PMAP Record		10-10
SL File Format		
Storage Management		
Entry Point Directory		
Typical Directory Entry		10-14
Code Segment Linkage Structure		
Reference Table Structure		
Reference Table (256 Maximum Entries)		
Code Segment With Patch Area		
PMAP Information		
PMAP Type Table		
PMAP Records		
Type 0		
Type 1		
Type 2	• • • •	10-21
CHAPTER 11 LOADER		
MPE Loader		11-1
Loader Segment Table Overview		
LST Overview		
XLST Overview		
Loader Segment Table Primary DB		
Directory Entries		11-3
Loader Cache		
Cache Data Segment Format		
Bucket Format		11-10
Loader Communication Table (LCT)	. <b></b>	11_11
LUAUCI CUIIIII UIII LAUIC (LCI)		11 11

Form Incoming to Loader (Load/Allocate Program)	-11
Form Incoming to Loader (Load/Allocate Procedure)	-12
Form Returned (No Error)	-13
Form Returned (Error Occurred)11-	-13
Logical Segment Transform Table (LSTT)	-14
Degioni Deginent Transform Table (DDTT)	-14
CHAPTER 12 PRIVATE VOLUMES / SERIAL DISC	
Mounted Volume Table (MVT AB)	2-1
Private Volume User Table (PVUSER))	2-4
Bind Names Data Segment	
Serial Disc Tables and Data Structures	
Data Record Format	
End of File Format	
Contiguous Block Format	
Hole Format	
Gap Table Format	
SDISC Extra Data Segments	
Serial Disc Organization	
56141 2 110 Organization	15
CHAPTER 13 I/O	
I/O Table Linkage	2_1
Device Reference Table (DRT)	
Driver Linkage Table (DLT)	
Logical-To-Physical Device Table (LPDT)	
Entry 0	
Typical Entry (Virtual Devices)	
Typical Entry (All Real Devices)	
Entry for Terminal-Like Devices	
Entry for Tape Drives	
Entry for Disc Drives	
Logical Device Table (LDT)	
Overview of Data Segment	
Zero Entry Format	
Typical Entry Format	
Logical Device Table Extension (LDTX)	
Overview of Data Segment	
Zero Entry	
Typical Entry	
Terminal Entry	
Serial or Foreign Disc Entry	
CIPER Entry	
System or Private Volume Disc Entry	
Device Class Table (DCT)	
Overview of Data Segment	
Header Entry Format	
Typical Entry Format	
Discussion	-16
Interrupt Linkage Table (ILT)	1 0

Device Information Table (DIT)					
DIT for HP-IB Systems					
DIT Terminology for HP-IB Systems					
Device Information Table (DIT) for CIPER					
DIT for Channel Devices		 		. 1	3-24
DIT For 7905/7906/7920/7925					
Error and Retry Information		 		1	3-28
CS 80 Disc Device Information Table (DIT)		 		1	3-29
DIT For 7970 Magnetic Tape		 		1	3-32
DIT for 7976 Magnetic Tape		 		1	3-34
Card Reader DIT		 		1	3-37
Card Reader DIT Field Definitions		 		1	3-38
Device Information Table for HP-IB Card Reader		 		1	3-39
2608 Line Printer DIT (HP-IB Systems)		 		1	3-41
2608 Line Printer Status		 		1	3-43
2631 or 2619A Line Printer DIT (for HP-IB Systems)		 		1	3-44
2680A/2688A DIT		 		1	3-46
I/O Status Block		 		1	3-48
Disc Request Table and Disc Requests		 		1	3-51
DISCREQTAB		 		1	13-51
Disc Request Table		 		1	l 3-52
Disc Request Table Entry 0 Format		 		1	13-52
Disc Request Element Format		 		1	l 3-53
IOQ Table Layout		 		1	l 3-56
I/O Queue Element (IOQ)		 	• 6	1	13-58
I/O System Status Returns		 		1	l3-60
I/O Queue Element for 7976A Magnetic Tape		 		1	l 3-62
I/O Queue Element (IOQ) for CIPER		 		1	l 3-64
HP-IB CIPER Physical Driver Request Codes					
CIPER Driver Return Status Codes					
2608 Line Printer I/O Queue Element (HP-IB Systems)		 		1	13-66
2608 Line Printer Request Codes		 		1	l 3-69
2619A & 2631 Line Printer IOQ Element (HP-IB Systems)		 		1	l 3-70
2619 Line Printer Request Codes		 		1	l 3-73
2631 Line Printer Request Codes (HP-IB Systems)					
I/O Queue Element For HP-IB Card Reader					
CS 80 Disc Request Queue Element (IOQ)		 		]	l 3-77
CS 80 Integrated Cartridge Tape Request		 		1	13-80
SBUF Table Layout		 		]	l 3-83
Table Element Allocation (SBUF)					
ICS Global		 			13-86
ICS Global Cells With Initial Values					
CS 80 Disc Interrupt Linkage Table (ILT)	• •	 	•	• •	13-90
CHAPTER 14 SPOOLING					
Input Device Directory/Output Device Directory		 			14-1
Overview of Table Structure		 	•		14-1
Entry 0 (Overall Table Definitions)		 			14-2

Typical Head Entry (4 words)																
Typical Subentry (%40 words)																
SPOOK Tape Format																
Label Record																14-6
File Directory																14-7
Device and Class Directory																14-7
Logical Device Entry																14-7
Device Class Entry																
Spoolfile Format																
Spoolfile Block Format																
Spoolfile Record Format																
User Labels Information																
osci Baccis information		• •	•	• •	• •	•	• •	•	• •	•	• •	•	•	• •	•	14-7
CHAPTER 15 UNIFIED COMMAND	LAN	GU	ΑG	E	(UN	NCI	.)									
Reply Information Table (RIT)																15-1
Message System General Description .																15-2
Message Catalog																
MAKECAT Program																
Message System CATALOG.PUB.SY																
Message Set Directory																
HELP Subsystem																
UDC Directory																
UDC's COMMAND.PUB.SYS																
CI Stack Definition																
Field Definitions																
Association DST Layout																
Issociation Dol Eagout	• • •		•	• •	• •	•	• •	•	• •	•	• •	•	•		• •	13-13
CHAPTER 16 SYSDUMP/INITIAL																
CONFDATA File																1.6 1
Popul O of CONEDATA Ella (CTA)	 DO)	• •	•	• •		•	• •	•	• •	•	• •	•	٠		•	16-1
Record 0 of CONFDATA File (CTA)	BUJ.	• •	٠	• •		•	• •	•	• •	•		•	•		•	16-1
Record 1 of CONFDATA File (CTA)	3)		•	• •		•		•	• •	•		•	•		•	16-2
DEVDATA.PUB.SYS																
Overview																
Parameter Record																
Driver Table																
SYSDUMP Format																
WSC Table Format																
Series 64/68 WCS Table Format																
Store Tape Format																
First Volume																
Subsequent Volumes																
End of Volume		٠.	•		٠.	•		•		•		•	•		. 1	16-13
CHAPTER 17 MISCELLANEOUS																
Labeled Tame College																
Labeled Tape Subsystem																
Tape Label Table																17-5

Volume Recognition				. 1	7-10
Opening a File				. 1	7-10
Reading and Writing Files				. 1	7-10
Closing Files				. 1	7-11
Store-Restore				. 1	7-11
Miscellaneous				. 1	7-11
Breakpoint Table				. 1	7-12
PCB Breakpoint Extension Table				. 1	7-13
Breakpoint Entry Table				. 1	7-13
Active Entry				. 1	7-14
Timer Request List (TRL)				. 1	7-17
MPE User Logging				. 1	7-19
General Design Overview				. 1	7-19
Hardware Environment.				. 1	7-19
Software Environment		•	•	1	7-19
Design Narrative		•	•	1	7-19
Error Recovery Description.	• •	• •	•	1	7-20
Design Structures			•	1	7-21
User Logging Table		•	•	1	7_21
User Logging Table	• •	• •	. •	. 1	7-21 7-27
User Logging Buffer	• •			. 1	7-27
User Logging Identifier Table	• •			. I	7-31
Logging Record Format			• •	. 1	7-41
Measurement Information Table	• •		• •	. 1	/-44
Message File Data Structures					18-1
File Structure					18-1
Block Structure					18 - 2
Record Format					18 - 2
Header Format					18 - 3
Message Access Control Block					18 - 3
MMSTAT Definitions					18-9
File System Basic IPC Definitions					
General Behavior					
Port Data Structures				. 1	8-13
Port Data Segment		•		. 1	8-13
Port With Two Outstanding Messages				. 1	8-13
Port Number				. 1	8-14
Port DST Number Array				. 1	8-14
Port Data Segment Global Area		٠		. 1	8-15
Port				. 1	8-16
Message Queue Entry (MQE)				. 1	8-1
File System Message Files				. 1	8-1'
Timer List Entry (TLE)				. 1	8-13
MMSTAT Definitions				. 1	8-18
CHAPTER 19 MPE MEMORY RESIDENT MESSAGE FACILITY					
					10
Overview of Facility					19-
Message Intrinsics					19-1

SENDMSG	- 1
PORTSTATUS	-2
RECEIVEMSG	-2
Supporting Data Structures	- 3
CHAPTER 20 MMSTATS EVENTS	
MMSTATS Catalog Index	_ 1
MMSTAT Event Group 0 (Memory Management Events)	
Event 0	
Event 1	
Event 2	
Event 4	
Event 5	
Event 6	
Event 7 (%7)	
Event 8 (%10)	-
MMSTAT Event Group 1 (Memory Manager)	
Event 12 (%14)	
Event 13 (%15)	
Event 14 (%16)	
Event 15 (%17)	
Event 16 (%20)	
Event 17 (%21)	
Event 18 (%22)	
MMSTAT Event Group 2	
Event -20 (-%24)	
Event -21 (%25)	
Event -23 (-%27)	
MMMSTAT Event Group 3	
MMSTAT Event Group 4 (Scheduling)	
Event 40 (%50)	16
MMMSTAT Event Group 5	
MMSTAT Event Group 6 (FILESYS)	
Event -60(%74)	
Event -61(%75)	
Event -60(%74)	
Event -61(%75)	
Event -62(%76)	
Event -63(%77)	
Event -64(%100)	
Event -65(%101)	
Event -66(%102)	24
Event -67(%103)	
Event -68(%104)	
Event -69 (%105)	25
MMSTAT Event Group 7 (FILESYS)	26
Event -70 (%106)	26
Event -71 (%107)	26
Event -72 (%110)	

Event -74 (%112)	20-27
Event -75 (%113)	
Event -76 (%114)	20-28
Event -77 (%115)	20-29
Event -78 (%116)	20-29
Event -79 (%117)	20-30
MMSTAT Event Group 8	20-31
Event -80 (%120)	20-31
Event -81 (%121)	20-31
Event 83 (%123)	20-32
Event 84 (%124)	20-33
Event 86 (%126)	20-33
Event 87 (%127)	20-33
Event 88 (%130)	20-34
Event 89 (%131)	20-34
MMSTAT Event Group 9 (Disc I/O Requests)	20-35
Event 90 (%132)	
Event -98 (%142)	
MMSTAT Event Group 10	
Event 100 (%144)	
Event 101 (%145)	
MMSTAT Event Group 11	
Event -110 (%156)	
Event -111 (%157)	
MMSTAT Event Group 12	
Event 120 (%170)	
Event 125 (%175)	
MMSTAT Event Group 13	
Event 139 (%213)	
MMSTAT Event Group 14 (CS/3000)	20-40
Event 140 (%214)	
Event 142 (%216)	
Event 144 (%220)	
Event 146 (%222)	
Event 147 (%223)	
Event 149 (%225)	20-43
MMSTAT Event Group 15 (CS/3000)	
Event 150 (%226)	
Event 152 (%230)	20-44
Event 153 (%231)	20-45
Event 154 (%232)	20-45
Event 155 (%233)	20-46
MMSTAT Event Group 16	20-47
Event 160 (%240)	20-47
MMSTAT Event Group 19	
Event 191 (%277)	
Event 192 (%300)	
Event 193 (%301)	20-50
Event 194 (%302)	20-51
Frank 105 (9202)	20 61

MMSTAT Event Group 20	.20-52
Event 200 (%310)	. 20-52
Event 201 (%311)	. 20-52
MMSTAT Event Group 21 Process Creations and Terminations	
Logical Process	.20-53
Logical Process Table	.20-53
Event -211 (%323)	.20-53
MMSTAT Event Group 22	. 20-54
Time Stamp of Event Trace Enable and Disable	. 20-54
Event 221 (%335)	. 20-54
Event 222 (%336)	. 20-55
Event -223 (-%337)	. 20-55
Event -224 -(%340)	. 20-56
Event -225 (-%341)	. 20-56
Event -226 (-%342)	. 20-56
Event -227 (-%343)	. 20-57
Event -228 (%344)	. 20-57
Event -229 (-%345)	. 20-57
MMSTAT Event Group 23 (Terminal I/O)	. 20-58
Event 230 (%346)	. 20-58
Event 231 (%347)	. 20-58
Event 232 (%350)	. 20-59
Event 233 (%351)	. 20-59
Event 234 (%352)	. 20-60
Event 235 (%353)	. 20-60
Event 236 (%354)	. 20-61
Event 237 (%355)	. 20-61
Event 238 (%356)	. 20-61
MMSTAT Event Group 24 (Power Fail)	. 20-62
Event 240 (%360)	. 20-62
CHAPTER 21 ROOTFILE LAYOUT	
General Rootfile Layout	. 21-1
Root File Label 0	. 21-2
Root File Labels 1 & 2	. 21-5
Root File Label 3	. 21-6
Root File- Next Label	. 21-7
Item/Set Read/Write Table Format	
Root File Record 0	
Root File Record 1	
Root File - Next Record	. 21-11
Data Set Control Blocks (DSCB) - General Layout	. 21-12
Data Set Control Block (Global Area)	. 21-13
Data Set Control Block (Item Numbers)	. 21-14
Data Set Control Block (Record Definition Item Displacement)	. 21-14
Data Set Control Block (Path Table)	. 21-15
General Data Set Layout	. 21-15
Data Set User Label 0	. 21-16
Data Set Records	21-16

# CHAPTER 22 DISC FREE SPACE MAP Disc Resident Data Structures 22-1 Bit Map 22-1 Descriptor Table (DT) 22-1 Virtual Memory Resident Data Structures 22-2 Disc Free Space Data Segment 22-2 CHAPTER 23 MPE DISC CACHING Disc Caching Overview 23-1 Disc Caching Tables Overview 23-4 Cache Directory Table 23-6 Header Entry 23-7 Device Entry 23-10 Mapped Domain Entry 23-13 Logical Disc Request Table 23-16 Logical Disc Request Entry 23-17

### **PREFACE**

This manual describes the internal table organization of the MPE V operating system, release G.00.00. The Tables Manual is an informational reference for the technically sophisticated user with Privilege Mode capability. We strongly discourage modifying the table structure because you may destroy the operating system. The following caution applies:

### CAUTION

The normal checks and limitations that apply to the standard MPE users are bypassed in Privileged Mode. It is possible for a Privileged Mode program to destroy file integrity including the MPE operating system software itself. Upon request Hewlett-Packard will investigate and attempt to resolve problems resulting from the use of Privileged Mode code. This service is available on a time and materials billing basis. However, Hewlett-Packard will not support, correct, or attend to any modifications of the MPE operating system software.

The table structure of MPE V is significantly expanded from MPE IV. The operating system reflected in the table structure is the Fundamental Operating Software (FOS) version of MPE V. Your table structure may look different depending on the applications and uses of your system.

The information is presented in several different formats. This reflects the combined knowledge of several divisions and groups within Hewlett-Packard. Instead of taking the time to consolidate all the various formats, we chose to release the information quickly.

We hope you will find this edition informative. Your comments and suggestions are welcome via the "Reader Comment Sheet" at the back of this manual.

### CHAPTER 1 MEMORY LAYOUT

### Fixed Low Memory (Series 44/48/64/68)

X	CSTB (BASE OF CST TABLE)**	DEC
11	KCSTB (POINTER TO CURRENT EXECUTING	<u>'</u> "  1
	PRDGRAM BLDCK)	
2	DSTB (BASE OF DST TABLE)**	[2
3	0	13
41	O CPCB (CURRENT PCB INDEK )**	4 >PCB REL
3	QI (INITIAL B TON ICC)	13
6	ZI (INITIAL Z FOR ICS)**	16
7	SYSTEM INTERRUPT MASK HORD**	7
1D	DRTBANK (BANK OF DRT TABLE)	18
11	DRIADDR (BASE OF DRI TABLE)	19
12	DBBANK (FOR INITIAL'S STACK) *	1D
13	DBBANK (FOR INITIAL'S STACK) *  DB (FOR INITIAL'S STACK) *	11
14		  12
15		112
16		
17		100
20		
21		  17
	TEMPLR (TEMP STORAGE OF LIMIT REG)+	
23	LR (SYSTEM CLOCK LIMIT REGISTER) **	19
24   /	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	//// 20

### Fixed Low Memory (Series 44/48/64/68) (Cont.)

25	TR (TIME	SINCE LAS	T SOFT T	IMER INTERR	PT)**  21
26	SCST	(SYSTEM C	LOCK STA	TUS)**	22
27	SCLC	(SYSTEM C	LOCK LAS	T COUNT)**	23
30-37					24-3

NOTE: All pointers are absolute addresses.

G.00.00 1- 2

### Memory Layout

### System Global Area

G.∞0.00 1- 1

OCTAL	1	2	3	4	5	6	7	8	9	0	1	1	3	1	5			NAME
0					SYS	GLO	8									-		
1				0	ST	BAS	Ε									c	ST	
2				0	ST	BAS	Ε									į ps	T	
3				P	СВ	BAS	Ε									PC	8	
4				SHA	PTA	8 8	ASE									si	LL	
5				1	00	BAS	Ė									I0	00	
6				SB	UF	BAS	Ε									  BI	JF	
7					IC	\$ 0	I									- u	cs	
10				LP	DT	BAS	E									ם	PDT	
11				SHO	ΝВ	ASE										- 31	101	
12				T	RL	BAS	E									 11	₹L	
13				JC	UT	BAS	E			•						- s:	(R	
14				S	IR	BAS	Ε									_ s	OCT	'AB
15	<b>-</b>			JPC	NT	BAS	Ε									ᆙ	PCN	IT
16				TB	UF	BAS	E									BL	JF	
17			D	ISC	RE	OUE	ST	BAS	E							DF	RO	
20	<b>-</b>															-		
21				Lī	K3 I	rĸ	EE	ΠEΠ	UKT	AD	UKE	.55						
22																-		
23					11	ΠĖ	UF	TH2	1 [	YCL	£					-		
24					R	ESE	RVE	D								-		
25				Br	eak	Ро	ınt	F1	ag							B	PTF	:
																1		

### Memory Layout

### System Global Area (Cont.)

		ı
26	VOSHTAB BASE	MATHEO
27	STATIC FENCE	
30	CURRENT CST BLOCK INDEX	СЅТВХ
31	MEASIO BASE	MEASIO
32	DISPLACEMENT TO CODE =@CST(O)-@DST(O)	OFC
33	DISPLACEMENT TO SHARABLE = @CST(LAST)-@OST(O)	OFS
34	Snon Index	
35	ABS ADDRESS (SYSOIT(8))	DITS
36	Reserved	SBANK
37	ABS ADR OF PMBC TABLE FOR LST/STT CHECKING	SBASE
40	RESERVED FOR INITIAL (VDSENTAY)	
41	RESERVED FOR INITIAL (VDSMAP)	
42	SRTTAB BASE	SRTTAB
43		SPECOHEAD
44	Number of Available Regions	HOLECOUNT
45	# PAGES IN LARGEST CURRENTLY AVAILABLE REGION	MAXAVAILREG
46	MAKE OVERLAY CANDIDATE INFORMATION	MOCINFO
47	NUMBER OF MEMORY BANKS CONFIGURED -1	NBANKS
50	SCHEOULER TO AWAKE MESSAGE	SCHEDTORWAKENSG
51	POINTER TO CSTBLK TABLE	CSTXBLCKPOINTER
52	AWAKE TO SCHEDULER MESSAGE	AWAKETOSCHEDMSG
53	HAIT TO SCHEDULER MESSAGE	
54	CURRENT ACTIVITY'S PRIORITY	CURACTPRI
		l

Пеног	~u t	200	114
HEHOL	, .	۳y∨	<b>u</b> 1

### Memory Layout

### System Global Rrea (Cont.)

	/55	BUSY TABLE POINTER	BUSY
	56	NERO TABLE POINTER	NERD
	57	TAIL TABLE POINTER	TAIL
	60	N OF SIO PROGRAMS EXECUTING	SIOCOUNT
	61	PRRITY ÊRROR FLAG (MEM PE)	PARITY
	62	Impeded queue head for message buffer (PIM)	IOMSGPIN
	63	I/O Message system error flags (0:1) - No SYSBUF avail for I/O error logging (1:1) - No SYSBUF for IOMESSAGE (GENMSG)	IOLOGQX
RESERVEO FOR I/O	64	# OF TERMINALS READING	ROCOUNT
SYSTEM	65	# OF TERMINALS WRITING	HATCOUNT
	66	DSET B	CRIO
	67	LAST TIMER	CRIO
	70		CRIO
	71	NIGHEST ORT NUMBER	NSYSORT
	72	POWERFRIL	POWERFAIL
	73	SYSTEM UP FLAG	SYSUP
	\74	SYS CONSOLE LOGICAL DEVICE NUMBER	CONSLOEV
1	75	COLO LORO COUNT	CLOROIO
	76	SMRREO FCB OST	SHFCBOST
	77	MONITORING FLAGS	!
FOR FILE	100		MAXSSECT

### System Global Area (Cont.)

	i	
102 	CURRENT # OF SPOOL XILOSECTORS	I Inumssect
103		
\104	N SECTOR/SPOOLFILE EXTENT	EXTSSECT
105	MAX CODE SEGMENT SIZE	ļ
106	MAX N OF CODE SEGMENTS/PROCESS	
107	MAX STACX SIZE (MAXOATA)	į
110	OEFAULT STACK SIZE	!
<b>1</b> 11	MAX EXTRA ORTR SEGMENT SIZE	ļ
112	MAX # EXTRA OATA SEGMENTS/PROCESS	!
113	OST number for MESSAGE buffers	ļ
114	UPORTE LEVEL	UPORTEL
115		FIXL
116		VERSION
117	DEFRULT CPU TIME LIMIT	İ
120	# OF SECONOS TO LOGON	į
121	JOBSYNCH BITS (13:3)	
122	EXTERNAL PLABEL OF INITIATE	<u> </u>
123	INTERNAL PLABEL OF INITIATE	]
124	MAXSYSDST	
125	MAXSYSCST	<u> </u>
126	Ldev for SL.PUB.SYS   NODA for SL.PUB.SYS	!
127	LOOR for St. PUB. SYS	!
130	(OIRECTORY)	!
131	(OISC MOORESS)	!
		1

G.00.00 1- 6

G.00.00 1- 5

Memory Layout

Memory Layout

### System Global Area (Cont.)

	132	SPOOLINDEX	
	/133	EXT LABEL FOR SNOWCOM	
	134		
DE052 150	135	CS IOWAIT PLABEL	
RESERVEO FOR CS	ļ136	CS FIX LEVEL	
	137	CS VERSION	
	\140	CCLOSE PLABEL	
	141	LOGICAL PROCESS TRBLE (PROGEN)	0
	142	<del></del> <del></del>	
	143	LOGICAL PROCESS TRBLE (UCOP)	2
	144	LOGICAL PROCESS TABLE (PFRIL)	3
	145	LOGICAL PROCESS TABLE (DEVAEC)	4
	146	LOGICAL PROCESS TABLE (ORUSG)	5
	147	LOGICAL PROCESS TABLE (STMSG)	6
	150	LOGICAL PROCESS TABLE (LOG)	7
	151	LOGICAL PROCESS TABLE (LOAD)	B
	152	LOGICAL PROCESS TABLE (IOMESSPROC)	9
	153	LOGICAL PROCESS TABLE (SYSIOPADC)	10
	154	LOGICAL PROCESS TABLE MEMLOGP	11
	155	EXTERNAL PLABEL OF "TERMINATE"	
	156	INTERNAL PLABEL OF "TERMINATE"	
	1		

System Global Area (Cont.)

	157	EXTERNAL PLASEL OF "COMMENDINTERP"	-
	160		į
	161		·
	162		·İ
	163		
	164		·
	165		1
			ļ
	166 -		.}
-	167	3 MORO	j
į	170	LOGGING	ļ
į	171	MASK	1
į	172	STATE  OSTN - BUFFER O	STATE:
ļ	173	STATE  OST# - BUFFER 1	O EMPTY
ļ	174	BUFFER LENGTH (SECTORS)	2 FULL
ļ	175	FREE AREA POINTER	1
DESERVED	176	FLAGX	
FOR	177	# RECORDS WRITTEN IN BUFFER O	
LOGGING !	200	N RECORDS WRITTEN IN BUFFER 1	
	201	FILE SIZE (BLOCKS) - 1ST HRLF	
	202	FILE SIZE (BLOCXS) - 2NO HRLF	
ļ	203	(LOG FILE SIZE)	
	204	(BLOCKS)	
	205	LOG FILE NUMBER (LOGFILENUM)	
	206	NUMBER OF LOGGING (BLOCKS WRITTEN (1ST HALF))	!
-	207	BLOCKS URITTEN [BLOCKS WRITTEN (2ND HRLF)]	
	-	**	1

						0 . 0 . 1 0 . (6. 4.)	
		System Global Area (Cont.)				System Global Area (Cont.)	
	210	(TOTAL # LOG RECORDS MISSED)		SEGMEN		CURRENT HORO COUNT	   KOSCOUNT
į	211	(OUE TO LOG FAILURE)		TRACE		BUFFER SIZE	   BUFFSIZE
į	į.	TOTAL# RECORDS MISSED - "JDB INITIRTION" LOSS			256	MRG TAPE LDEV	i I LDEV
LOCCTNC	- 1			<u>i</u>	- 257	TRRCE SEGMENT EXTERNAL LABEL	TLRBEL
LOGGING I	ĺ	TOTAL# RECORDS MISSEO - "JOB TERMINATION" LOSS			260	STHON	i i i i i i i i i i i i i i i i i i i
	I	OPERRTOR COHSOLE JOBSESSION \$ RT STARTUP			j-		į
	215  216				261	MERSINFOTROPTR	   
	217				262	MERSUREMENT STRTISTICS CLASS MRSK	GCLASSENRBLEO
	1	HRPPING FIRMURE FLRG (NON-ZERO=MPE V/E UCODE)			263	CLASS O STRTISTICS BRNK NUMBER	MERSSTRTXOSBRNK
		BRNK AND ROORESS OF MAPPING OST ( INITIRLIZED BY DISPRICHER OURING LAUNCHING R PROCESS)			264	CLRSS D STRTISTICS ROORESS	MERSSTSTXOSBRSE 
	223	TOTAL SEGMENT NUMBER OF CURRENT PROCESS			265	SCRH POINT	ļ
	224	TOTAL FREE PHYSICAL CST ENTRIES			266    -		1
	225	HERO OF FREE PHYSICAL CST LINK			267	MERSFLAGS	** 
	226				270   -  -		1
	227				271	INDEX OF PCB RT HERO OF DISPATCHING Q	SYSOISQNERO
	247	RESERVEO			272	INDEX OF PCB RT TRIL OF DISPRTCHING Q	SYSOISPOTRIL
	1	HOLE LIST NEAD (BRNK)	NLHEAO		273	OST # OF COT TRBLE (DISC CACHING)	į
	250	HOLE LIST HERD (RDDRESS)	l Maleno	KERN	274	BRNK # OF THE COT TRBLE (OISC CRCHING)	
	251		MITOTI	, NEAR	275	ADDRESS OF COT TRBLE (DISC CRCHING)	
	<b>2</b> 52	HOLE LIST TAIL (BRNK)	MLTRIL		276	MELP LOGICAL DEVICE HUMBER	ļ
	253 <b> </b>	HOLE LIST TAIL (RODRESS)			277	CURRENT LOGON OST	OSTLOGON
					300	(STOP)	
					301 302	(BITS) (see p. 2-15) # PROCESS ENTRIES	
		G.∞0.∞0 1- 9				6.00.00 1- 10	
A. L. W.							
-4-i <del>2</del> -i		1- 9	Menory Layout	Menory (	ayout	1- 10	
			Menory Layout	Menory (	ayout		
	304	1- 9  System Global Rrea (Cont.)	Menory Layout	Menory (	Ī	1- 10  System Global Area (Cont.)	·
	304	1- 9  System Global Rrea (Cont.)  DEVREC PIN   2	Menory Layout	Nenory (	335	1- 10  System Global Area (Cont.)  DSCHECK PLABEL	-!    -  -
	305	1- 9    System Global Rrea (Cont.)	Menory Layout	Пеногу (	335 336	1- 10  System Global Area (Cont.)  DSCHECK PLABEL  GSDPEN PLABEL	-! 
	305 306	1- 9    System Global Rrea (Cont.)	Menory Layout	Пеногу (	335   336   337	1- 10  System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL	-! 
PRICESS	305 306 307	1- 9  System Global Rrea (Cont.)  DEVREC PIN   2  X20  UCOP PIN   D  X20	Menory Layout	Nenory (	335 336 337 340	1- 10  System Global Area (Cont.)  DSCHECK PLRBEL  OSDPEN PLRBEL  OSCLOSE PLRBEL  MRNRGEURITE CONV. PLRBEL	-   -  -  -  -  -  -
PROCESS STOP TRBLE	305 306 307 310	1- 9    System Global Rrea (Cont.)	Menory Layout	Menory (	335 336 337 340 341	1- 10  System Global Area (Cont.)  DSCHECK PLRBEL  OSDPEN PLRBEL  OSCLOSE PLRBEL  HRNRGEURITE CONV. PLRBEL  CONSDSLINE' PLRBEL	-
STOP	305 306 307 310 311	1- 9  System Global Rrea (Cont.)  DEVREC PIN   2  X20  UCOP PIN   D  X20  LOG PIN   1  X20	Menory Layout	Menory (	335 336 337 340 3411 342	1- 10  System Global Area (Cont.)  DSCHECK PLABEL  OSCUSE PLABEL  OSCUSE PLABEL  MRNRGEWRITE CONV. PLABEL  CONSDSLINE' PLABEL  CKREHOTE PLABEL	-  
STOP	305 306 307 310 311 312	1- 9	Menory Layout	Menory (	335 336 337 340 341 342 343	1- 10  System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  MRNRGEWRITE CONV. PLABEL  CONSDSLINE' PLABEL  CXREMOTE PLABEL  CXREMOTE PLABEL	
STOP	305 306 307 310 311 312 313	1- 9	Menory Layout	<b>Меногу I</b>	335 336 337 340 341 342 343	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  MRARGEURITE CONV. PLABEL  CONSOSLINE' PLABEL  CXRENOTE PLABEL  CXRENOTE PLABEL  CXRET PLABEL  CXRET PLABEL	
STOP	305 306 307 310 311 312	1- 9	Memory Layout	<b>Меногу (</b>	335 336 337 340 341 342 343 344 345	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  MRNRGEURITE CONV. PLABEL  CONSDSLINE' PLABEL  CKREHOTE PLABEL  CKREHOTE PLABEL  CKREF PLABEL  OSIANGE PLABEL	-   
STOP	305 306 307 310 311 312 313	System Global Rrea (Cont.)    DEVREC PIN   2	Menory Layout	Menory (	335 336 337 340 341 342 343 344 345 346	System Global Area (Cont.)  DSCHECK PLABEL  OSOPEN PLABEL  OSCLOSE PLABEL  CONSOSLINE' PLABEL  CXREMOTE PLABEL  CXREMOTE PLABEL  CXRET PLABEL  OSITRE PLABEL  OSITRE PLABEL  OSITRE PLABEL	-1
STOP	305 306 307 310 311 312 313 314	1- 9	Menory Layout	Menory (	335 336 337 340 341 342 343 344 345 346	System Global Area (Cont.)  DSCHECK PLRBEL  OSDPEN PLRBEL  OSCLOSE PLRBEL  CONSDSLIME' PLRBEL  CKREMOTE PLRBEL  CKREMOTE PLRBEL  CKRER PLRBEL  OSTINGE PLRBEL  OSTINGE PLRBEL  OSTINGE PLRBEL  OSTINGE PLRBEL	-!    -
STOP	305 306 307 310 311 312 313 314	1- 9    System Global Rrea (Cont.)	Menory Layout	Nenory (	335 336 337 340 341 342 343 343 345 346 347	System Global Area (Cont.)  DSCHECK PLRBEL  OSDPEN PLRBEL  OSCLOSE PLRBEL  CONSDSLIME' PLRBEL  CKREMOTE PLRBEL  CKREMOTE PLRBEL  CKRER PLRBEL  OSTINGE PLRBEL  OSTINGE PLRBEL  OSTINGE PLRBEL  OSTINGE PLRBEL	-!    -
STOP	305 306 307 310 311 312 313 314 315	System Global Rrea (Cont.)    DEVREC PIN   2	Menory Layout	Menory (	335 336 337 340 341 342 343 343 345 346 347	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  CONSOSLINE' PLABEL  CKRENOTE PLABEL  CKRENOTE PLABEL  CKREN PLABEL  OSTARGE PLABEL  OSTARGE PLABEL  OSTARGE PLABEL  OSTARGE PLABEL  OFFAULT LABEL TYPE   TRPE LBL AUTO REC FUN  SYSDB PTR TO TERM INIT CHNL PGM (\$30/33 ONLY)  MP! IS	-     -    SOFTOEATH FLA
STOP	305 306 307 310 311 312 313 314 315 316	1- 9	Menory Layout	Пеногу (	335 336 337 340 341 342 343 344 345 346 347 350	System Global Area (Cont.)  DSCHECK PLRBEL  OSDPEN PLRBEL  OSCLOSE PLRBEL  CONSDSLIME' PLRBEL  CKREMOTE PLRBEL  CKREMOTE PLRBEL  CKRER PLRBEL  OSTINGE PLRBEL  OSTINGE PLRBEL  OSTINGE PLRBEL  OSTINGE PLRBEL	-     -    SOFTOEATH FLA
STOP	305 306 307 310 311 312 313 314 315 316 317	1- 9	Menory Layout	Пеногу (	335 336 337 340 341 342 343 344 345 346 347 350	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  CONSOSLINE' PLABEL  CXREMOTE PLABEL  CXREMOTE PLABEL  CXREMOTE PLABEL  CXRET PLABEL  OSINGE PLABEL  OSTRICE PLABEL  SYSDB PTR TO TERM INIT CHNL PCH (\$30/33 OMLY)  TP]  LRST CYCLE OURRITON	-     -    SOFTOEATH FLA
STOP	305 306 307 310 311 312 313 314 315 316 317 - 320 321	1- 9	Menory Layout	Пеногу (	335 336 337 340 341 342 343 344 345 346 347 350 351	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  CONSDINE' PLABEL  CARRENTE PLABEL  CXREHOTE PLABEL  CXREHOTE PLABEL  CXRET PLABEL  OSINGE PLABEL  OSINGE PLABEL  OSINGE PLABEL  OSINGE PLABEL  OSTROE PLABEL  OSTROE PLABEL  OFFRUIT LABEL TYPE   TRPE LBL AUTO REC FUN  SYSDB PTR TO TERM INIT CHAL PCH (\$30/33 ONLY)  MP  IS  LAST CYCLE DURRTION	-     -    SOFTOEATH FLA
STOP	305 306 307 310 311 312 313 314 315 316 317 - 320 321 322 323	1- 9	Memory Layout	Пеногу (	335 336 337 340 341 342 343 344 345 346 347 350 351 352	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  CONSOSLINE' PLABEL  CONSOSLINE' PLABEL  CXREMOTE PLABEL  CXREMOTE PLABEL  CXRET PLABEL  OSTINGE PLABEL  OSTINGE PLABEL  OSTINGE PLABEL  OSTINGE PLABEL  OFFAULT LABEL TYPE   TRPE LBL AUTO REC FUN  SYSDB PTR TO TERM INIT CHAL PEM (\$30/33 OMLY)  MP!   S  LAST CYCLE OURRITON	-     -    SOFTOEATH FLA
STOP TRBLE	305 306 307 310 311 312 313 314 315 316 317 - 320 321 322 323	System Global Rrea (Cont.)    DEVREC PIN   2	Menory Layout	Menory (	335 336 337 340 341 342 343 344 345 346 347 350 351 352 353	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  INNARGEWRITE CONV. PLABEL  CONSDSLINE' PLABEL  CXREHOTE PLABEL  CXREHOTE PLABEL  CXREF PLABEL  OSINGE PLABEL  OSINGE PLABEL  OSINGE PLABEL  OSINGE PLABEL  OSTROGE  OSTR	-     -    SOFTOEATH FLA
STOP TRBLE	305 306 307 310 311 312 313 314 315 316 317 320 321 322 323 324	System Global Rrea (Cont.)    DEVREC PIN   2	Menory Layout	Menory (	335 336 337 340 341 342 343 344 345 346 347 350 351 352 353 354	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  CONSOSLINE: PLABEL  CONSOSLINE: PLABEL  CKRENOTE PLABEL  CKRENOTE PLABEL  CKREN PLABEL  OSTARGE PLABEL  OSTARGE PLABEL  OFFAULT LABEL TYPE   TRPE LBL AUTO REC FUN  SYSDB PTR TO TERM INIT CHNL PGM (\$30/33 ONLY)  MP! IS  LAST CYCLE OURRITON  CYCLE THRESHOLD	-   
STOP TRBLE	305 306 307 310 311 312 313 314 315 316 317 - 320 321 322 323 324 325	System Global Rrea (Cont.)    DEVREC PIN	Menory Layout	Menory I	335 336 337 340 341 342 343 344 345 346 350 351 352 353 354 355 356	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  MRNRGEWRITE CONV. PLABEL  CONSDSLINE' PLABEL  CXRENOTE PLABEL  CXRENOTE PLABEL  CXRET PLABEL  OSITRGE PLABEL  OSTRGE PLABEL  OFFAULT LABEL TYPE   TAPE LBL AUTO REC FUN  SYSDB PTR TO TERM INIT CHNL PGM (530/33 ONLY)  MP] IS  LAST CYCLE OURRITON  CYCLE THRESHOLD  BUG CATCH EMABLE CELL  HONLTOR BUFFER   TIMESTRMP	- I SOFTOEATH FLA
STOP TRBLE	305 306 307 310 311 312 313 314 315 316 327 320 321 322 323 324 325 326	System Global Rrea (Cont.)    DEVREC PIN	Menory Layout	Menory I	3351 3360 3471 3421 343 3443 345 346 347 350 351 352 353 354 355 356 357 360	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  CONSOSLINE' PLABEL  CXREMOTE PLABEL  CXREMOTE PLABEL  CXREMOTE PLABEL  CXREMOTE PLABEL  OSTINGE PLABEL  OSTINGE PLABEL  OSTINGE PLABEL  OFFAULT LABEL TYPE   TEPE LBL AUTO REC FUN  SYSDB PTR TO TERM INIT CHNL PGH (530/33 OMLY)  TP  IS  LAST CYCLE OURRITON  CYCLE THRESHOLD  BUG CATCH ENABLE CELL  HONITOR BUFFER   TIMESTRMP	- I SOFTOEATH FLA
STOP TRBLE	305 306 307 310 311 312 313 314 315 316 327 321 322 323 324 325 326 327 330	System Global Rrea (Cont.)    DEVREC PIN   2	LAST	Menory I	3351 3361 3471 3422 3431 3443 3451 350 351 352 353 354 355 356 357 360 361	System Global Area (Cont.)  DISCHECK PLRBEL  OSDPEN PLRBEL  OSCLOSE PLRBEL  MRNRGEWRITE CONV. PLRBEL  CONSOSLINE' PLRBEL  CXREMOTE PLRBEL  CXREMOTE PLRBEL  CXREMOTE PLRBEL  OSITRGE PLRBEL  OSITRGE PLRBEL  OSTRGE OSTRGE PLRBEL  Bank of last memory word	- SOFTOEATH FLB - MEH PRESSURE - I - MONBUFTD - MONBUFT1
STOP TRBLE	305 306 307 310 311 312 313 314 315 316 322 323 324 325 326 327 330 331	System Global Rrea (Cont.)    DEVREC PIN   2	LRST	Menory I	335 336 337 340 341 342 343 344 345 355 354 355 356 357 360 361 362	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  MRNRGEWRITE CONV. PLABEL  CONSDSLINE, PLABEL  CXREHOTE PLABEL  CXREHOTE PLABEL  CXREF PLABEL  OSINRGE PLABEL  OSINRGE PLABEL  OSTRINGE PLABEL  TYPE   TRPE LBL RUTO REC FUN  SYSDB PTR TO TERM INIT CHNL PCH (\$30/33 ONLY)  HP! IS  LAST CYCLE OURRITON  CYCLE THRESHOLD  BUG CATCH ENABLE CELL  HONITOR BUFFER   TIMESTRIP  OSBREAK PLABEL  Bank of last Henory WORD  Base of last Henory WORD	SOFTOEATH FLA HEM PRESSURE  - HONBUFTD - HONBUFT1 - LAST MEMORY
STOP TRBLE	305 306 307 310 311 312 313 314 315 316 327 321 322 323 324 325 326 327 330	System Global Rrea (Cont.)    DEVREC PIN   2	LAST   1   1   1   1   1   1   1   1   1	Menory I	3351 3361 3471 3422 3431 3443 3451 350 351 352 353 354 355 356 357 360 361	System Global Area (Cont.)  DSCHECK PLABEL  OSDPEN PLABEL  OSCLOSE PLABEL  CONSDSLINE' PLABEL  CKREHOTE PLABEL  CKREHOTE PLABEL  CKRER PLABEL  OSINGE PLABEL  OSINGE PLABEL  OSINGE PLABEL  OSINGE PLABEL  OSTROE PLABEL  TO TERM INIT CHAL PCH (\$30/33 OHLY)  MP]  IS  LAST CYCLE THRESHOLD  BUG CATCH ENABLE CELL  MONITOR BUFFER   TIMESTRIP  OSSREAK PLABEL  Bank of last memory word  Base of last memory word  PVPROC PIM	SOFTOEATH FLA HEM PRESSURE  - HONBUFTD - HONBUFT1 - LAST MEMORY

G.00.00 1- 11

Menory Layout

### SysGlob Extension

1200 words long; Pointer found at SysOB + 1377

	1	
X 0	SURP QUEUE DELRY (*100MS)	SUAPOCELAY
1	BANK OF FIAST REGION IN LINKED MEMORY	FIRST
2	BASE OF FIRST REGION IN LINKED MEMORY	REGION
3	GARBAGE COLLECTION ENABLE FLAG	GARBCOLLENAB
4	MOVE THRESHOLD (IN PAGES, FOR GARB COLL)	MOVETHRESH
5	MAIN MEMORY PAGE SIZE (IN WOROS)	
6	VOS PAGE SIZE	
7	LAST MAKE ROOM TIME	
10		
11	MEMORY PRESSUAE OURATION THRESHOLD	
12	RESERVEO FOR NATIVE LANGUAGE SUPPORT	
13	RESERVED FOR NATIVE LANGURGE SUPPORT	
14	BRUO RATE OF THE SYSTEM CONSOLE	
15	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
16	PLABEL FOR REMOTE'MPE	
56	1	
1		
57	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
60	PLABEL USERLOG (EXTERNAL)	
61	PLABEL USERLOG (INTEAHRE)	
62	PLABEL RECLOG (EXTERNAL)	

6.00.00 1- 14

G.00.00 1- 13

Menory Layout

SysGlob Extension (Cont.)

63	PLABEL RECLOG (INTERNAL)	ļ
64	PLABEL RESTART (EXTERNAL)	1
65	PLABEL AESTART (INTERNAL)	
66	PMBC LOW CORE BANK # (USER)	
67	PMBC LOW CORE ADDRESS (USER)	
70	RESERVEO FOR IMAGE	
71	RESERVEO FOR MEASIO 12  MIOCHT	*
72	LOROER CRCHE SEGMENT NUMBER	į
73	PLABEL 3270 (EXTERNAL)	<u> </u>
74	VERSION	!
75	UPDATE	
76	FIX	!
77	COUNT OF TRPE CONTROLLERS USING MEASIO	<u> </u>
100	PORT ORTR SEGMENT NUMBEA	
101	RESERVEO FOR SECONO PORT DATA SEGMENT	
102	SYSTEM FPMRP OPTION FLRG	SYSFPMAP
103 104 105 106 107 110	GLOBAL ALLOW MASK	
111	RESERVEO	
117		
120	SYS PORT PROCESS PCB RELATIVE INDEX	
121	GLOBAL RFT OST NUMBER	

Memory Layout

SysGlob Extension (Cont.)

122	INITIAL/PROGEN COMM. OSEG NUMBER
123	
127	CURRENTLY UNASSIGNED
130	(OS, NETWORK MGMT, APPLICATION SERVICES)
131	<del></del>
132	
133	<del></del> 
134	<del></del> 
135	
136	<del></del>
137	
140	
141	
142	
143	
144	
145	RESERVEO FOR SPL
146	PRTN FLOW
147	
i	RHRLYZER
150	
151   	CURRENTLY UNRSSIGNED
ı	
200	
	·

\* HIOCHT = MERSIOCOUNT (3 BITS)
\*\* MERSFLAGS (15:1) = 1 ==> MONITOR ENABLED

(14:1) = 1 ==> BUFFER FLIP/FLOP (13:1) = 1 ==> EOT ON MONITOR TRPE

### SYSDB Words

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		A	ddr	ess							I	В	ank		
1															1

Address is the whole word with "Bank" masked out to 00000.

Systems that have MPE V/E microcode (all 6% systems, 4% systems with new boards) can have a non-zero bank number. Systems running pre-MPE V/E microcode can only use bank 0, therefore the pointer will look like:

	2								15 
	Rd	dre	SS						j
I	 			 	 	 	 	 	1

SysG1ob	Mord	Defi	ini	tions

ADORESS	NAME	FUNCTION
DB+55	BUSY	- SYSDB relative pointer to BUSY TABLE for I/O resources
DB+56	HEAD	- SYSDB relative pointer to table containing head pointers to I/O resource queues
08+57	TAIL	<ul> <li>SYSDB relative pointer to table containing head pointers to tail of I/O resource queues</li> </ul>
DB+60	SID COUNT	- Number of I/D Programs currently executing
DB+72	POWER FAIL	- O-no power fail 1-system disc recovery 2-all other disc recovery
		3-all other device recovery
0B+73	SYSUP	- System is up and operable
0B+74	CONSLDEVN	- System comsole logical device number
0B+400	CPU NUMBER	- Sét when system aborts

6.00.00 1- 17

UPPER LIMIT->DEVICE COMMANDS

ABORTJOB

ALLON ALTFILE ALTJDB BAEAKJDB DELETE DISALLON JDBFENCE

LIMIT STOPSPOOL SUSPENDSPOOL OUTFENCE RECALL RESUMEJOB RESUMESPOOL STREAMS CONSDLE

13 14 15

Memory Layout

JOBSYNCH job synchronization via jobsynch (sysglob+121(B))

(13:1) - JDBSAEADY - set by DEVREC & MORGUE (via procedure STRATDEVICE) indicating a ready job. This prevents UCDP from going to a wait state when a job is just made ready.

(15:1) - DEVFREED - set by DEALLDCATE when device count goes to 0.

NDTE - Both bits above used for synchronization of job-made-ready or devicefreed when UCDP is running.

(14:1) - JDBSWATITNG- set by UCDP just before waiting if any job is waiting for list device. Signals DERLLOCATE to awake UCDP when a device is freed.

### Allow Mask Format

The Allow mask for MPE V is expanded to six words. There is a mask in each user's JIT and in the SYSGLOB area. The Allow mask contains enough bits for a one-to-one correspondence to every present DPERATDR type command, or any future OPERATOR command. When a user is ALLOWed any DPERATOR command or ASSOCIATED to a device (which will use DPERATDR type commands) then the corresponding bit(s) in the mask in that user's JIT for that command is set. If the ALLOW or RSSOCIATE was done on a global scale, then the bit(s) in the mask of the SYSGLOB area is/are updated.

The following EQUATEs define the mask bit for each operator command.

The first set of commands define the operator commands dealing with devices.

When adding a new command to this set of EQURTEs, be sure to add a corresponding move statement in LDGIMRGE, even if the command will not be logged.

	Hord	<u>Bit</u>	#
ABORTID	0	0	0
ACCEPT	0	1	1
DDIIN	٥	2	2
GIVE	0	3	3
HEADDEF	0	4	1 2 3 4 5
HERDON	Ó	5	5
REFUSE	Ó	6	6
REPLY	٥	7	7
STRATSPOOL	0	8	8
TAKE	0	9	9
UP	Ó	10	10
MPLINE	Ó	11	11
DSCONTROL	Ö	12	12

6.00.00 1- 18

Allow Mask (Cont.)

Menory Layout

Memory Layout

	<u>Hord</u>	<u>Bit</u>	#
HARN HELCORE HIDN HIDF YHOUNT LHOUNT LDISHOUNT LDISHOUNT OUNLORO OUNLORO HIDENBLE HIOOISABLE LOG FOREIGN INF SHOUCOH OPENQ SHUTT	112222222222222222222222222222222222222	14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
OISCAPS	3	2	48

### Logging Related Locations

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

172		-
or	[STATE]	DST #
477	1	

STATE = 0 if respective buffer empty 1 if respective buffer is current 2 if respective buffer is full

FLAGX

SF = 1 if soft failure
HF = 1 if hard failure
BUF = 0 if current log buffer is buffer 0
= 1 if current log buffer is buffer 1
SL = 1 to indicate a suitch in log buffers (from 0 to 1 or from 1 to 0)
SO = 1 to indicate shutdown in progress

### Process Stop List General Layout

SYSDB	
300	STDP BITS REPRESENTING WHICH PROCESSES TO STOP ON "SHUTDOWN"
	# PROCESS ENTRIES
	///////////////////////////////////////
	1ST PROCESS ENTRY
	2ND PADCESS ENTRY
	:
317	LAST PROCESS ENTRY
317	

### Entry Format

0	1											14	15 
	F	ADC	ESS	PI	N H	'					BIT		
				р	AOC	ESS	UR	IT	ST	RTE	 	 	

### Preassigned Entries

entry #	process	stop bit #
1	devrec	2
2	ucop	Ō
3	ucop Iog	1

G.00.00 1- 21

### Initial Memory Allocation

This section is a description of the nethod used by INITIAL to allocate memory for MPE tables and code segments in MPE V/E. All memory allocated by INITIAL is permanently allocated. All non-core resident code and data is put on disc before exiting INITIAL.

At the most basic level INITIAL will try to build memory to look exactly as diagrammed below. There are, however, several ways in which to deviate from this structure. Before going into the sources of these deviations, it is necessary to point out which portions of memory are used by INITIAL during the restart and therefore cannot be used by MPE until INITIAL has finished.

Before INITIAL begins to allocate any memory space, it relocates ite core resident code, its code segment swapping area and its stack to the highest configured memory space. Additionally, it uses the last X326 words of bank O in series &x nachines for its I/O buffer area and temporary code segment table. After INITIAL has built all of core resident MPE (tables and code), it builds the disc resident MPE tables. Since some of the disc resident tables may be too large to be built in INITIAL's stack, these tables are built in unused memory space. Therefore, in addition to the memory space required for INITIAL'e code, INITIAL's stack and core resident MPE, there must be enough space left in which to build the largest of the disc resident tables.

For Series 6x nachines with the MPE V/E firmware, INITIRL will build the tables with ">" signs by them out of Bank O if necessary. For all other tables, INITIRL will essentially build memory in the order shown below. There may be an unused fragment of memory between the DRI's and the system global area which INITIRL will fill with the snaller tables. Neither the tables narked with an asterisk nor the code segments will ever be put in this area. NOTE: INITIRL will build all tables on 32-word boundaries.

If the system being built by INITIRL is configured with 128K words or 160K words of memory them INITIRL's stack will be in bank 1 (the code also on a 128K word memory size). If INITIRL is occupying part of bank 1 and the space is needed for a core resident NPE code segment or to build a disc resident table them INITIRL will print the error message "EAROR M350 OUT OF MEMORY".

Except for the exceptions stated above, for every allocation of memory INTIRL will first try to allocate any renaining space between the DRI's and SYSOB. It will then try the next available space in bank 0, then the next available space in bank 1. If it were necessary it could continue searching until all all banks were checked for available space.

Inmediately before exiting INIVIRI, INIVIRI lays down all the nemory region headere and trailers as shown below. For any one bank of memory there will only be one block of core resident NPE, regardlees of its contents. The only block of core resident NPE that does not have a reserved region global header is an bank O. It does have the reserved region global trailer though. Before placing any code outside bank O the first 24 words of every bank (except bank D) is reserved for the region global header.

G.00.00 1- 22

Memory Layout

Bank O

Lau Core Henory	
>DRT	  (Only on 64/68 if Pri-   vilege Mode Bounds
System Global area	Checking is enabled.)
Firmware area	
SYSGLOB Extension	
DST/CST/CSTX	
ICS	
PMBC	(Dnly for 64/68 if Pri- vilege Mode Bounds
ILT/DIT	Checking is enabled.)
DLT	
Resource Tables	
EST Block	!
>Memory Measurement Info	
VDSM Table	
Job Process Count	
> PRI/SEC MSR	į
>PC8	į
> Swap Table (Sii)	į
>Special Aequest Table	į
>Job Cutoff Table	ļ
>Timer Request List	
>System Buffers	
>LPDT	
>IOQ	
>SIR	
>MON Table	İ

Memory Layout

Bank O (Cont.)

Core Resident CST's in CST order Reserved Region Global Trailer Rvailable Region Global Neader Rvailable Memory Rvailable Region Global Trailer

NOTE: The > neans these tables can move out of Bank D if necessary.

Bank 1

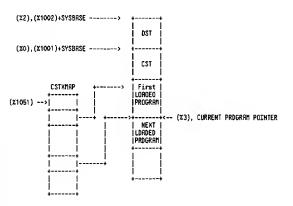
Reserved Region Global Neader Core Resident CST's and tables marked with ">" the didn't fit in BANK D Reserved Region Global Trailer

### Memory Management

### CHAPTER 2 HENGRY MANAGEMENT TABLES

### Segment Table Structure

The current location and state of each data segment and loaded code segment is maintained in the Segment Table. This table is partitioned into three separate tables as shoun in Figure 2-1. The partitions are based on the segment classes: a segment is a data segment, a segment is a system SL segment, or a segment is part of a program. The structure and format of each partition is described in the following.



Overall ST Structure

Memory Management

### Pointers and DST #'s of Segment Table Components

### i. DST

% 2 absolute address of entry D of the DST. %1002 sysbase relative index of entry O of DST. DST number 2 is the DST Table dst #.

### ii. CST

Z O absolute address of entry O of System SL. X1001 sysbase relative index of entry O of System SL. X1032 displacement from OSI base of entry O of System SL (i.e. <code>@CST(last) - @DST(D) = OFS</code>). DSI number 4 is the CSTX Table DST M.

### iii. CSTX

% 1 absolute address of entry 0 of current program. X1033 displacement from OST base to first CSTM entry SL. OST number 4 is the CSTM Table OST #.

### iv. CSTXMAP

<code>X1051</code> sysbase relative index of entry D of CSTXMRP. DST number 43 (X72) is CSTXMRP Table DST  $\mbox{\tt W}.$ 

G.00.00 2- 2

G.00.00 2- 1

Memory Management

Standard Object Identifier Format

O 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

TYPE | CSTBLK OBJECT NUMBER

DBJIDENTIFIER(D).(D:4) ==> TYPE = 0 Dbject is a Data segment = 1 Dbject is an SL segment = 2 Dbject is a Program segment = 3 Dbject is a Cache Oomain

### DST Entry Formats

DST/CST Entry O Format

5
-1
- 1
-1
İ
-i
i
-i
i
-i

Memory Management

DST General Entry Format

Case (i) OST Entry for a Present Data Segment

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
HORO O	IN O IN SIZE/4	FIRMINFD
WORD 1		FLAGS
NORO 2	BANK	MABANK
WDRD 3	BASE	MMBASE

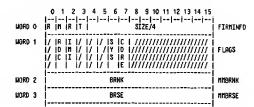
Case (ii) DST Entry for an Absent Data Segment

unan	٨	 	) 	2  -	2 : -	3 4 -1	1 ! -	5 I	5 ; -	7 8 -	9 10 11 12 13 14 1	5 -    FIRMINFD
NUNU	٧	1"	IV	In.	١.					317	/4	LIKITHIA
HORO	1	IC V			S  T  K 	M  0  0	F  W  I  P	S  Y  S 	IC IO IA IE	#   0  	VHRLLOC	FLAGS
HORO	2			LDE	V f	•				I	HOOR	HOOR
NORO	3	-						U	DDA			-    LOOA -
	HORO	HORO O HORO 1 HORO 2 HORO 3	HORO 0   R HORO 1   O   C   V 	HORO 0   A   O   O   O   O   O   O   O   O   O	HORO 0	WORO O	HORO 0	WORD 0   R   O   R   I   S   H   F	WORD 0   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   O	WORD 0   R   O   R     O   R     O   R     O   R     O   R   O   R   O   R   O   O	WORD 0   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   A   O   O	HORO 1   0   R   I   S   M   F   S   C   H

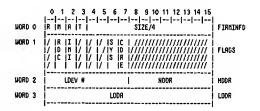
### **CST Entry Formats**

### CST General Entry Fornat

Case (i) CST Entry for a Present SL Segment or CSTX Segment



CRSE (ii) CST Entry For Rn Rbsent Segment SL or CSTX Segment



Case (iii) DST/CST Free Entry

Z100000
TRBLE RELATIVE OFFSET TO NEXT FREE ENTRY
TRBLE RELATIVE DFFSET TO PREVIOUS FREE ENTRY

Refer to the Logical Segment Table Format in Chapter 11 for more information on XCST.

6.00.00 2- 5

### Henory Hanagement

### ST Entry Field Descriptions

R = 1 ==> segment absent

N = 1 ==> segment privileged

R = 1 ==> segment privileged

R = 1 ==> segment has been referenced

I = 1 ==> segment is being traced

DCV = 1 ==> disc copy is valid

STK = 1 ==> segment is a stack

MOD = 1 ==> a segment nodification (exp., contr.) is pending

FUTP = 1 ==> a forced write of this segment is in progress

VMPRECENT = # of virtual nemory pages allocated to this segment

MOC = 1 ==> segment is recoverable overlay candidate

INI = 1 ==> segment is in notion

SYS = 1 ==> segment is a system segment

CDRE = 1 ==> segment is core resident

WD = 1 ==> urite disabled

### CSTBLK Format

CSTBLK(0)		*
*	NUMBER OF ENTRIES IN TRBLE	*
1		*
*	RHY UNRSSIGNED ENTRY = -1	*
2		*
*	RNY RSSIGNED ENTRY > ¢	*
3		*
*	REMAINING CSTBLK TRBLE ENTRIES	*

The table is initialized to minus one in each entry. When selected, the entry is replaced by a DST-relative index to the entry MO of the CST extension block. This ie the the overhead entry For the associated program.

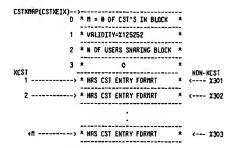
G.00.00 2- 6

### Memory Management

### Program Blocks and the CSTXMRP

Since programs can be dynamically loaded and unloaded, the segment table must be kept packed or Fragmentation would occur. Thus, the block of ST entries for a program segment begins at an ST entry number that changes if a program which was loaded before it gets unloaded. To manage this dynamic etructure, an auxiliary structure, the CSTMTRP is used. R program is identified by its index, CSTMTRT, into this map. The program's current beginning physical ST entry number is equal to equal to CSTMTP (CSTMEIX).

Entry Format - CST Extension Block



The value of CSTKEIX is established when a CST extension block is allocated. This index into the array CSTKMRP is naintained in the PCB of each process sharing the block.

### Memory Management

### Fixed DST Entry Rasignments

DCTRL	1	DECIMEL	TRBLE NRME
0		0	
1	CST	1	CST
2	DST	2	DST
3	PCB	3	PCB
4	CSTX	4	CSTX
5	SYSTEM GLOBRL RRER	5	SYS
6	CORE	6	CDRE
7	ICS	7	ICS
10	SYSTEM BUFFERS	8	SBUF
11	UCOP REQUEST QUEUE	9	UCRQ
12	PROCESS-PROCESS COMMUNICATION TRBLE	10	PPCDN
13	I/D QUEUE	11	IDQ
14	TERMINRL BUFFERS	12	TBUF
15	LDGICAL-PHYSICAL DEVICE TABLE	13	LPDT
16	LOGICRL DEVICE TRBLE	14	LDT
17	DRIVER LINKRGE TRBLE	15	DLT
50	I/D RESDURCE TRBLES	16	BUSY, NERD, TRIL
21	SECONDRRY MSG TABLE	17	SECMSGTAB
22	LORDER SEGMENT TRBLE	18	LST
23	TIMER REQUEST LIST	19	TAL
24	DIRECTORY	20	DDS

### Memory Management

### DST (Cont.)

-			
DCTAL		DECIMAL	TABLE NAME
25	DIAECTORY SPACE	21	
26	RIN TABLE	22	AIN
27	SHRPTABLE (SLL)	23	SURPTRB
30	JDB PADCESS COUNT	24	JPCNT
31	JDB MASTER TABLE	25	JHRT
32	TAPE LABEL TABLE	26 	VOD
33	LOG TABLE	27	LOGTRB
34	REPLY INFORMATION   TRBLE	28	RIT
35	VOLUME TABLE	29	VTAB
36	BREAKPOINT TABLE	30	STOP
37	LOG BUFFER1	31	
40	LOG BUFFER2	32	
41	LOG IO TABLE	33	LIOTRB
42	ASSOCIATE TABLE	34	
43	CST BLOCK	35	CSTBLK
44	JOB CUTOFF TABLE	36	JCUT
45	SYSTEM JIT	37	SJIT
46	SPECIRL REQ TABLE	38	SRT
47	VIRTURL DISC SPACE I MRNRGEMENT TABLE	1   39 	VDSHTAB
50	DEVICE CLASS TRBLE	40	DEVCLASS
51	Reserved Kernel	41	
		1	

Menory Management

### DST (Cont.)

DĈTAL	l	DECINAL	TABLE NAME
52	ILT	42	ILT
53	SIA TABLE	43	SIA
54	FMRVT	44	FNAVT
55	INPUT DEVICE DIRECT	45	IDD
56	OUTPUT DEVICE DIRECT	46	DDD
57	WELCOME MESSAGE #1	47	LOGONDSTN1
60	HELCOME MESSAGE #2	48	LOGONDSTN2
61	CS DRTR SEGMENT	49	CSTAB
62	PROCESS-JOB CROSS REFERENCE	50	PJXAEF
63	SYSTEM JOT	51	TOLZYZ
64	COMMAND LOGON OST	52	CILOGOST
65	MOUNTED VOL. SET TRBLE	53	MVTA8
66	PRI.VOL. USER TABLE	54	PVUSER
67	RESERVED KEANEL	55	
70	DISC REQUEST TABLE	56	DISCREQTAB
71	MSG HAABOR TABLE	57	MSGHARBTAB
72	PRIMARY MESSAGE TRBLE	58	PAINNSGTAB
73	MEASUREMENT INFO TABLE	59	MEASINFOTAB
74	FIRST FREE DST	60 	

G.00.00 2- 9

G.00.00 2- 10

### Menory Management

### Swap Tables

The SURPTRB is a core resident menory nanagement table used to keep track of the locality lists of the competing processes. The PCB entry for a process has a SURPTRB relative pointer to the header entry for the process.

SWAPTER 0ST# = 23 (X27)

%1004 System table pointer to SWAPTRB entry D.

NOTE: The number of entries configured will be 3 greater than the number configured via SYSDUMP. (Entry 0 consumes 3 entries).

### SWAPTAB Entry 0 Format

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	1
D		D
1	ENTRY SIZE (6)	1
2	# AVAILABLE ENTRIES	2
3	TABLE RELATIVE INDEX OF FIRST FACE ENTRY	3
4	TABLE AELATIVE INDEX OF LAST FREE ENTAY	4
5	HIGH WATER MARK	5
6	# PRIMARY ENTAIES (D)	6
7	NEAD OF IMPEDED QUEUE (PCB RELATIVE)	7
8	TAIL OF IMPEDED QUEUE (PCB RELATIVE)	10
9	# CUARENTLY IMPEDED PROCESSES	11
10	MAX # DF IMPEDED PROCESSES	12
11	CUMULATIVE # OF IMPEDED PADCESSES	13
12	<del></del>	14
	:	
17	{   	21

### Menory Management

### SMRPTAB Unassigned Entry Format

	D 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
٥	
1	TABLE RELATIVE INDEX OF NEXT FREE ENTRY
2	TABLE RELATIVE INDEX OF PREV. FAEE ENTRY
3	D
4	D
5	0
	1

Rn assigned entry in the swaptab is a process' SLL header or a member of a process' SLL. These formats are now described.

### Segment Locality Lists (SLL)

The system maintains for each process a segment locality list (SLL) of the segments belonging to that process' current working set. The process' SLL consists of a header and a list of entries. The header and list entries are taken from the SMAPIRE.

R process' SLL is located via the process' PCB entry. PCB01 contains the SLL relative index of the process' SLL header.





G.00.00 2- 13

Memory Management

### SLL Neader Format

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
0		SCHEOTD10nSG
1	TRBLE RELRTIVE INDEX OF FIRST ENTRY IN LIST	FIRSTINX
2	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
3	TRBLE RELATIVE INDEX OF MEHORY REQUEST ENTRY	HEMRE QINX
4	# ENTRIES IN PROCESS' SLL	SEGCDUNT
5	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	

- SLL(SLLHERDINX+O)
  .(1:1) SWREQ, SHAP Required Flag
  .(2:1) HRSMEN, Nas Memory Flag
  .(3:1) INTIDC, Intilalize locality to minimum
  .(4:1) PRRIIM, Process partially swapped in
  .(5:1) STRIOW, Start shap over Flag
  .(6:1) SWIP, Swap In Progress Flag
  .(8:8) IDCNT, Segment read completions until awake

G.00.00 2- 14

### Menory Management

### SLL List Entry Format

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
0	PCB RELATIVE INCEX OF THE NEXT IMPECED PIN	MEXTIMPPIN
1	TRBLE RELRTIVE INDEX TO NEXT ENTRY IN LIST	NEXTINX
2	TABLE RELATIVE INDEX TO PREV. ENTRY IN LIST	PREVINX
3	- OBJECT IDENTIFIER -	SLL'OBJOESC
4	- GBSECT IDENTIFIER -	SLL'OBJHUM
5		SLL'FLAGS

SLL(SLLINX+O) NEXTIMPPIN, next make present deferred queue PCB Index

SLL(SLLINX+1) NEXTINX, next SLL entry

SLL(SLLINX+2) PREVINX, previous SLL entry

SLL(SLLINX+3) SLL'OBJDESC, 1st word of object identifier

SLL(SLLINX+4) SLL'OBJNUM, 2nd word of object identifier

SLL(SLLINX+5)

- 5)

  (0:1) MRPSEG, process' CST napping segment (LSII)
  (1:1) STK, process' stack entry
  (2:1) DISCIDSEG, disc I/O pending on this segment
  (3:1) LDCKED, segment locked in memory
  (4:1) BLKLK, request for blocked lock
  (5:1) FROZE, segment frozen in memory
  (6:1) SLLINI, process queued for this segment
  (7:1) IDSS, loss this entry
  (8:1) FRZREQ, request segment to be frozen
  (9:1) LKRCQ, request to lock segment in memory
  (10:1) DCCNIFLEG.
  (10:1) PREFETCHCDUNT,

NDTE:

The Suap Table will be configured with at least turce the number of configured PCBs.

G.00.00 2-15

Memory Management

### Special Request Table

Used for passing data segment size change info and for keeping a list of devices waiting for a segment to arrive in memory.

%1042 - SRT relative index to entry H O %1043 - SRT relative index to the head of the queue

NOTE: The number of entries configured will be 3 greater than the number configured via SYSDUMP. (Entry #0 consumes 3 entries).

### SRT Entry O Format

0	N ENTRIES CONFIGURED
1	ENTRY SIZE (6)
2	# AVRILABLE ENTRIES
3	TRBLE REL. INDEX OF 1ST FREE ENTRY
4	TRBLE REL. INDEX OF LAST FREE ENTRY
5	NIGH WRTER MRRX
6	# PRIMARY ENTRIES
7	HERD OF IMPEDED QUEUE (PCB REL.)
8	TAIL OF IMPEDED QUEUE (PCB REL.)
9	N CURRENTLY IMPEDED PROCESSES
10	N MAXIMUM IMPEDED PROCESSES
11	CUMULATIVE N DF IMPEDED PROCESSES
12	
	<u>.</u>
	i :
17	     <b></b>

### Menory Management

The following entry format is for data segment size changes:

٥	NEXT ENTRY FOR DATA SEGMENTS
1	- OBJECT IDENTIFIER -
2	NEW DRTH SEGMENT SIZE
4	READ DISPLACEMENT
5	MOVE COUNT

The following is the format for devices maiting on a segment: (The region header for the segment contains an SRT relative index to this entry. If more that 5 devices are maiting on this segment, another entry will be linked to this entry.)

0	NEXT ENTRY OF QUEUED DEVS ON SEG
1	IDQINX
2	I IOQINX
3	IODINX
4	IOQINX
· 5	IDQINX
•	

The number of primary configured entries will be equal to the total number of LDEVs configured. The number of secondary entries will be configured to be at least the same as the number of PCBs configured. Data segment change entries are secondary type, while devices queued entries will be primary entries.

G.00.00 2- 17

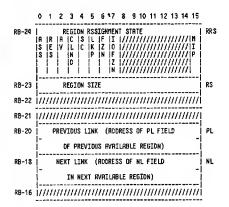
Memory Management

Header length = 24 Trailer length = 4

Global Region Trailer

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
         PREVIOUS TRAILER SUBREGION SIZE
                                        PTSS
RB-27
         PREVIOUS TRAILER REGION STRTE
                                        PTRAS
RB-26
    PREVIOUS TRRILER REGION SIZE
R8-25
```

### Global Region Neader (Available Regions)



Menory Management

### Main Memory Region Headers and Trailers

Main memory is partitioned into regions. Each region is in one of three states: available, reserved, or assigned.

An available region is available for consumption by the free space allocation mechanism. An available region consists of neighboring subregions, each of which is either a hole or an overlay candidate. An available region is linked into the available region list.

A reserved region is a main memory region which is in the transition state from available to assigned. A reserved region has been cleaned, and there is a pending disc read of a segment into the region.

Assigned regions are occupied by present segments. Available and reserved regions consist of one or more adjacent subregions. Region headers and trailers are partitioned into global and local components. The global region header/trailer is only valid for the first/last subregion in regions consisting of more than one subregion.

The region headers and trailers of available, reserved, and assigned regions contain the state and control information pertaining to the current or planned contents of the region.

Cache domains are another form of assigned regions and are designated as such in the subregion header. If the cache domain is "napped" - has I/O pending against it - then the object identifier will have a non-zero value in the second word of the segment identifier field. If the second word of the segment identifier field is zero, then this region is a cache domain that is unnapped. (Refer to Chapter 23 for further information regarding Disc Caching.)

G.00.00 2-18

Memory Management

Subregion Header (Available Regions)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
	SUBREGION ASSIGNMENT STATE   C   R   R	SAS
RB-14	SUBREGION SIZE	\$\$
R8-13	V   SUBREGION DISPLACEMENT IN MAIN MEM. PAGES	SD
RB-12	URITE REQUEST POINTER	UREOP
RB-11	- DRJECT IDENTIFIER -	OBJIDENT
RB-9	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
RB-8	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
RB-7	LDEV   HODA	HODA
R8-6	Low Order Disk Address	LODR
R8-5	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
R8-4	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
R8-3	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
RB-2	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
RB-1	\mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm	

### Global Region Header (Reserved Regions)

		D	1	2	3	4	5	6	7	8	9	10	11	1	2	13	1	4	15	
RÐ	i	A S	IR IS	IA		C	K	IGN  F  Z  N	II D	1//	TA1							/I /I /I /I	H I P	RAS
RO-	-23			R	EGI	OH	SIZ	E												RS
RB-	-22			DN	GO	IHG	I/	0 0	OUN	IT										IOCNT
		S P R O C E S	IX ID IS IA B	IO IN IG ID IN IG	NIT IU IE IS IE IG IR	IAT	ION  E  N  P  R  O  U  E	IG IR IR IB IB IG IE	SSF IN IS IG IA IB ID IR IT	GE IR IE IP IB IB	IS IS IR IR						////////////////////////////////////	71/1/1/1/1/1/1	H S G V R L I D	INITHSG INITIRFO
R8		N S G P R	jr je	8  L  K  D	CD IS IC IN IE ID	MPL  I  O  H  A  I	ETI  M	1//	MES	111	7//							///////////////////////////////////////	N S G V A L I D	COMPRISG
RB-	-18		MAK	E P	RES	ENT	DE	FER	RED	00	EUE	(	PCB	ī	MD	EX	)			MPOLINK
RB-	-17			R	ELE	ASE	PR	GE	CDU	NT										PAGECNT
RB-	-16	S	PEC	IAL	RE	OUE	ST	TAE	E	PTR	( (	RT	TA		E	RE	L)			SPECRECTABPTR

Memory Management

### Subregion Header (Reserved Regions)

	D 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
R8-15	SUBREGION ASSIGNMENT STRTE  C  r  r  r	SRS
RB-14	SUBAEGION SIZE	22
RB-13	V + SUBREGION DISPLACEMENT IN MAIR MEM. PAGES	<b>\$</b> D
RB-12	WRITE REQUEST POINTER	UREOP
RB-11	- DBJECT IDENTIFIER -	DBJIDEHT
RB-9	FREEZE COURT   LOCK COUNT	LKFZCNT
RB-8	HRITE DISRBLE COUHT   I/D FROZEN COUNT	HDIDFZCHT
RB-7	LOEV   HIGH ORDER DISC ADDRESS	HDOA
RB-6	LOW ORDER DISC RODRESS	LDDA
RB-5	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
RB-4	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
RB-3	TIME OF	RRRTIME
	- ARRIVAL	
RB-1	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
	· .	

G.00.00 2- 21

G.00.00 2- 22

### Menory Management

### Subregion Header (Cached Regions)

```
I SAS
                 SUBREGIOH SIZE
RB-14
                                              SS
      V | SUBREGION DISPLACEMENT IN MAIN MEM. PAGES
RB-13
                                              SO
       WRITE REQUEST POINTER
                                              WREQP
RB-12
RB-11
                                              OBJICENT
                OBJECT IDENTIFIER
         PREVIOUS CACHED REGION (RODRESS OF PD
RB-9
                                              PO
            FIELD OF PREVIOUS CACHED REGION)
           LDEV | HIGH ORDER DISC ADDRESS
RB-7
                                              HODR
            LOW ORDER DISC HODRESS
RB-6
                                              LODA
RB-5
         NEXT CRCNED REGION (RODRESS OF ND
                                              NO
            FIELD OF NEXT CRCHED REGIDN)
               TIME OF
                                              ARRITHE
RB-3
         ARRIYAL
DISC ROORESS CSL(8)
                                             CRCDRDISP
RB-1
```

### Henory Hanagement

### Region Header and Trailer Field Descriptions

RRS,	Region Rssignment State .(0:1) Region Rssigned Flag .(1:1) Region Reserved Flag .(2:1) Region Rvailable Flag .(3:1) Region Cleaned Flag .(3:1) Region Locked Flag .(5:1) Region Locked Flag .(5:1) Region Frozen Flag .(7:1) Region I/O Frozen Flag .(8:1) LSIT segment .(8:1) LSIT segment .(8:5) Bot used .(15:1) Blocked Lock Migration in Progress Flag
IOCRT,	Dn-Going I/O Count = N of on-going I/O's in the region which must complete before the initiation message can be processed.
INITHSG,	Initiation Message .(0:1) Message Frocessed Toggle Switch .(1:1) Message Externally Disabled Flag .(2:1) Message Externally Disabled Flag .(2:1) Message On-going I/O Disabled Flag .(3:1) Deue Segnent Read Disc Request Flag .(4:1) Incore Hove Request Flag .(5:1) Expansion Request Flag .(5:1) Expansion Request Flag .(7:1) Message Mborted Flag .(7:1) Message Mborted Flag .(9:1) Ok to start completion flag .(9:1) Ok to seart completion flag .(10:5) Not used .(15:1) Message Valid Flag
INITINFD,	Initiation Message Ruxiliary Information = DRG relative index of segment read disc request if IMITMSC. OREADREC=1 or = +/- Displacement to initiation message for moves and expansions.
COMPMSG.	Completion Message

### Completion Message

.(D:1)	Message Processed Toggle Switch
.(1:1)	Segment Modification Required
.(2:1)	Block Lock Request
.(3:1)	Send Scheduler A Message
. (4:1)	Ruaken A Device
. (5:1)	Message Rhorted
. (6:9)	Available
. (15:1)	Message Valid Flag

### Menory Management

**MPQLINK** PCB relative index of the NERD of the make present

PRGECNT,

Release Page Count
=# of extra pages to release before processing initiation nessage.

SPECREQIABPIA, A Special Request Table relative index to the list of devices queued on this segment.

SAS,

Subregion Rssignment State .(0:1) Cached region .(1:1) Referenced .(2:1) Recover Dverlay Candidate .(13:3) I/D Status fron region fetch

SS. Subregion Size

SD,

Subregion Displacement .(0:1) Displacement Count Valid Flag .(1:15) # Pages to Base of Region

UREOP.

Write Request Pointer
= DRQ Relative Index of Oisc Write Request when the
Data Segment in the Subregion is in Motion Out
When the region belongs to a cashed domain which
is napped (i. e. OBJIDENT = 30000/non zero number)
this word is non zero. If the cashed domain is not
napped UREQP is zero.

DBJIDENT, Dbject Identifier- has standard object identifier format

Lock and freeze count .(0:8) Number of times region has been frozen .(8:8) Number of times region has been locked

WDIDFZCNT, Iofreeze count
.(0:8) Not used
.(8:8) Number of times region has been iofrozen

For regions belonging to cashed domains, the above two words contain the absolute address of the PD field in the previous region belonging to a cashed domain.

NDDA. Nigh order disc address in virtual memory of this region

Low order disc address in virtual memory of this region  $% \left( \left\langle n\right\rangle \right) =\left\langle n\right\rangle \left\langle n\right$ LODA,

ND.

Next cashed domain link for cashed domain regions only. Contains the absolute address of the ND field of the next cashed region.( 2 words )  $\,$ 

G.00.00 2- 25

### Menory Management

ARATIME. Arrival time, contains the time at which the segment contained in the region became present

CREDADISP

Valid only for regions containing a cashed domain, this word represents the disc address ( in one word ) of the segment contained in the region. This word which exists in each member of a linked list of cashed domains, is used as the target word during the LLSM instruction.

### Space Allocation Structures

As of MPE V/P and V/E, one doubly linked list structure is used instead of the multiple lists ordered by size as in MPE TV. Sysglob locations X250 through X253 contain the respective head and tail (bank & address) of the available region list. These four words have in essence replaced the ARSBM and RAL data structures in MPE TV. Memory allocation and deallocation is handled through PUIDNARL and TAKEDFFRAL. The search for an available region of the desired size is done via the LLSM instruction. The format of the list is the following:

Sysglob Z250 & Z251 points to the absolute address of the MEXT LINK field (two words) in the first available region on the list. The NEXT LINK field in the first available region points to the absolute address of the NEXT LINK field in the second available region and so on. It is worth mentioning that in addition to having a NEXT LINK field, each available region also contains a PREVIOUS LINK pointer, which makes management of the list both easier and faster.

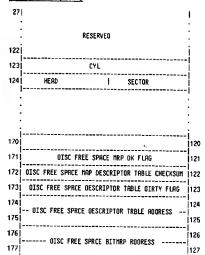
G.00.00 2- 26

Disc Layout Disc Layout System Disc Layout (Cont.) CHAPTER 3 DISC LAYOUT -----|SECTOR # SECTOR # System Disc Layout SECTOR # SECTOR # DISC COLD LORO INFORMATION TRBLE 28 DISC LABEL DISC COLD LOAD INFORMATION TRBLE 29 DEFECTIVE TRACKS/SECTOR TABLE DISC COLD LORD INFORMATION TABLE 30 36 COLO LORO CHANNEL PROGRAM FOR HP-IB SYSDUMP/INITIAL COMMUNICATION RECORD 31 37 MEN OUMP CHRNNEL PROGRAM FOR MP-IB 32 DISC COLD LORD INFO. TABLE EXT. DISC COLD LORD INFO. TRBLE EXT. 33 41 CODE FOR
INITIAL PROGRAMS
"BOOTSTRAP"
SEGMENT VARIABLE LENGTH | FOLLOHS | IMMEDIRTELY | RFTER | BODTSTRRP | SEGMENT LDW CORE (CST POINTER, QI, ZI, POINTER) TEMPORRRY CST (INITIRL PROGRRM) INTERNAL INTERRUPT WALTS BDOTSTRRP STRCK REMAINOER OF SIO COLD LORD PROGRAM G.00.00 3- 2 G.00.00 3- 1 Oisc Layout Oisc Layout Disc Label (Sector O of Disc) System Disc Layout (Cont.) System Volume 4 5 6 7 8 9 10 11 12 13 14 15 --|--|--|--|--|--|--|--| CONTROL ORDER SYSDB <<CYL/RRC #>> X130/131 SYSTEM DIRECTORY READ ORDER

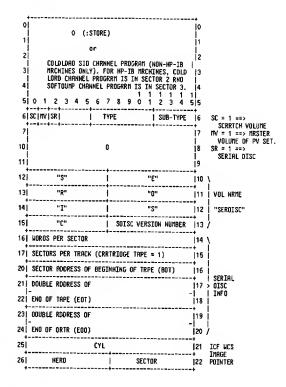
NOTE: INITIAL
TRIES TO
RLLOCRTE
OIRECTLY RFTER
THE FREE SPACE
MRP. NOUEVER,
THIS MRY
VARY DEPENDING
ON DELETED
OR RERSSIGNED
TRRCKS VIRTURL MEMORY RRER INITIAL PROGRAM SEGMENTS (EXCEPT BOOTSTRAP SEG) SYSTEM FILES (FROM COLD LORO TAPE) VOLUME TABLE INITIAL PROGREM STACK REMAINING INITIAL CODE SEGMENTS USER FILES

DISC BOOTSTRAP SIO PROGRAM (SYSTEM DISC ONLY) <<mem address>> Words 0-5 contain the ascil string "SYSTEM DISC" for SIO JUMP OROER << nen rdoress>> HP-IB Systems 6 ////////// DISC TYPE |DISCSUBTYPE 6 COLD LDAD ID "3" "0" IF WORD X11 CONTAINS R "1" A FORMER SYSTEM VOLUME HRS BEEN SCRATCHED. "0" "0" 12 13 VOLUME NAME 12 14 13 15 UNUSED ICF NCS 25 IMRGE POINTER SECTOR HEAD

System Volume (Cont.)



Serial Volume



G.00.00 3- 5

G.00.00 3- 6

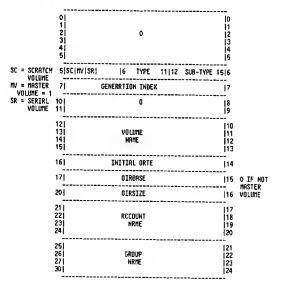
Oisc Layout

Serial Volume (Cont.)

+			+
27			123
122	RESERVED FOR F	UTURE HCS	82
123	CYL		183
1241	HEAO	SECTOR	84

Disc Layout

Master Volume



Disc Layout Disc Lavout Master Volume (Cont.) Slave Volume 31 | 32 | 33 | 34 | |25 |26 |27 HERDER SC = SCRATCH 5 | VOLUME | VOLUME = 0 | SR = SERIAL | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLUME | VOLU 129 130 6[SC[MVISR] | 16 TYPE 11,12 Ty8-TYPE 15[6 |31 |32 |33 |34 GENERATION INDEX VOLUME NAME YOLUME ENTRY O 10] 11] 43| 135 12| 13| 14| 15| 44| SUB-TYPE 136 45 113 137 161 INITIAL DATE VOLUNE 17] 20] 15 116 |17 |18 |19 |20 ACCOUNT NAME 251 261 271 301 21 22 123 124 170 Disc Free Space map OK flag 121 31 | 32 | 33 | 34 | 25 26 27 28 DISC FREE SPACE DESCRIPTOR TABLE CHECKSUM 1122 172 NAME SET 173 DISC FREE SPACE DESCRIPTOR TABLE DIRTY FLAG 123 124 DISC FREE SPACE DESCRIPTOR TABLE ADDRESS 175 176 DISC FREE SPACE BITHAP ADDRESS

> G.00.00 3-9

G.00.00 3- 10

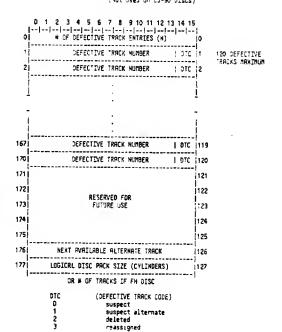
Disc Layout

Slave Volume (Cont.)



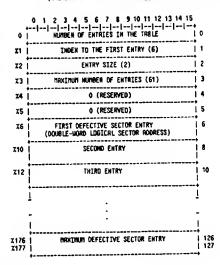
Disc Layout

Defective Imacks Table (Sector 1 of Disc)
(Not Uses On ES-30 Discs)



NOTE: The situation where there are two entries for the same track, n, one having a DTC of O (suspect) and the other naving a DTC 3 (reassigned) results from a situation where the disc driver could not "read" (unreadable) the address of the particular track.

### Defective Sector Table (DSCT -- Sector 1 of Disc) (the DSCT exists on device type 3 (CS-80) disce)

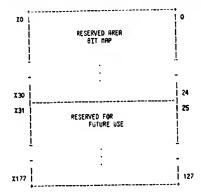


Unlike the DTT, entries in the DSCT are not permanent. Once a suspect sector to handled by INTITAL or VINIT, its entry is removed from the table. Thus, this table contains only unprocessed suspect sectors.

### Reserved Area 81t Hap (Sector 4 of the System Disc)

The first 400 sectors of the system disc are reserved for Initial'e use. This area contains permanent data structures for the boot. It is also used as a temporary storage area for data during sparing. Bill other system volumes and private volumes reserve only the first 10 sectors of the disc. They do not have a reserved area bit map.

The bit map contains 1 bit per sector. A '1' means the sector is free.



G.00.00 3- 13

6.00.00 3- 14

### Disc Layout

### Disc Cold Load Information Table (Sectors 28-30)

9	AGTALCH IR LUGIT TW DIGHTLEAN	FREFTR >
1	POINTER TO TEMPORARY CST INFO	TCSTPTR
2	W OF ENTRIES TO READ ON DISC COLD LOAD	MREAD
3	W OF CODE SEGMENTS IN INITIAL	MAIC21.
4	INITIAL'S OB VALUE	INITOB
5		INITOL
6		INITZ
7	INITIAL'S Q VALUE	INITO
8		IMITS
9	SYSDISC TYPE   SUBTYPE	DISCTST
10	COLD 1090 IO	נסוס,וסשס,נט,
11		LOG'FILE'HUM'
12	AFFERDAL ATER	DIRADA
13	ADDRESS	J. J. J. J. J. J. J. J. J. J. J. J. J. J
14	LOEV 1 VIRTURE HENDRY	VIRNENADOR
15	DISC ROORESS	VIRGINOU
16	N LOG PROCS	ļ
17		
18		ATMADR
19	DISC ROORESS	nanava.
20	OIRECTORY SIZE	DIRSECT
21	MSECTORS IN VIRTUAL MEMORY REGION OF LDEY 1	SECTORS IN LINEVIVI
22		
2	RIN THOLE SIZE	AINSECT
24	# OF STHE	#INS
-		÷1

Disc Layout

		GRIMS TL=Tape cold load
	TL RL AY	LDRO MODE RL=Reload
	NIGHEST VOL #   # OF VOLUMES	RY=recovery H'VDL'
	DISC COLD LORD ENTRY POINT	DISCENTRY
	SYSTEM DISC DRT NUMBER	SYSDISCORT
	JOB MASTER TABLE DTSC ADORESS	JHATLOC .
	IDD DISC ADDRESS	100L0C
	QOO DISC ROORESS	200FDC
	WELCOME MESSAGE (DST 47 10) DISC ADDRESS	LDGOHLDC1
	UELCOME MESSAGE (DST 48 10) DTSC ADDRESS	LOGONLOC2
     	LOG ID RODRESS	 
   	LOG TRO ADDRESS	
-	roe to size	
- 	LDG THE SIZE	į

## Disc Cold Load Information Table (Cont.)

	SIZE IN WORDS		FAEFTR+0	<]
       	MEMORY ADDRESS	*DRIVER TABLE		
! !	DISC ADDRESS			
   	SIZE IN HORDS		FAEFTR+5	
	MEMORY ADDRESS	*CTRBO		
     	DISC ADDRESS			
   	SIZE IN WDRDS		FAEFTR+10	
i   	MEMORY ADDRESS	*CTAB		
	DISC ADDRESS			
 	SIZE IN WORDS		FAEFTR+15	
	MEMORY ADDRESS	TION SUB-   SYSTEM DRIVER TABLE		
 	DISC ADDRESS			
	SIZE IN WORDS	* COMMUNICA-	FREFTR+20	
 	MEMORY ADDRESS	TION SUB-   SYSTEM   DEFINITION  TABLE		
]   	DISC ADDRESS			
•		1		

## Disc Cold Load Information Table (Cont.)

SIZE IN WORDS		   FAEFTR+25
MEMORY ADDRESS	COMMUNICA- SUBSYSTEM TABLE	
DISC ADDRESS		 
SIZE IN HORDS		FREFTR+30
MEMORY ADDRESS	LOGICAL- PNYSICAL DEVICE TABLE	
DISC ADDRESS		
SIZE IN WORDS		FAEFTR+35
MENORY ADDRESS	LOGICAL- DEVICE TABLE	
DISC ADDRESS		
SIZE IN HORDS	i	FREFTR+40
MEMORY ADDRESS	DEVICE CLASS TABLE	
DISC ADDRESS		
SIZE IN WORDS		FREFTR+45
MEMORY ADDRESS	VOLUME TABLE	
DISC ADDRESS		

G.00,00 3- 18

Disc Layout

Disc Layout

## Disc Cold Load Information Table (Cont.)

G.00.00 3- 17

		1
SIZE IN HORDS		FAEFTR+50
MEMORY ADDRESS	LOGICAL DEVICE TABLE EXTENSION	
DISC ADDRESS		
STACK SIZE		FREFTR+55
MEMORY ADDRESS	INITIAL'S STACK	
DISC ADDRESS		
SIZE IN HORDS		FREFTR+60
MEMORY ADDRESS	DEVICE CLASS TABLE HEADER	
DISC ADDRESS		
SIZE IN HORDS		FAEFTR+65
MEMORY ADDRESS	TERMINAL DESCRIPTOR TABLE	
DISC ADDRESS		
SEGMENT SIZE		FAEFTR+70
MEHORY ADDRESS	INITIAL/   SYSDUMP   COMMUNICATION   RECORD	
DISC ADDRESS		

Disc Cold Load Information Table (Cont.)

SEGMENT SIZE		FAEFTR+75
MEMORY ADDRESS	INITIAL'S SEGMENTS	
DISC ADDRESS		
(MORE SEGMENTS OF INITIAL) ININ		  -  -
TNITTOL D		ı

## INITIAL Program CST Map

LOGICAL <u>CST#</u>	PHYSICAL CST#	SEGMENT HAME
9 1 2 3 4 5 6 7 10 11 12 13 14 15 16 17 20 21	1 2 3 4 5 6 7 10 11 12 13 14 15 16 17 20 21 22 23	ININ  BOOTSTRAP  > core resident  RESIDENT /  RENSEGH

\*code segment swapping starts at completion of MRINSEG1  $\,$ 

## SYSOUMP/Initial Communication Record

D	MIT VERSION
1	MIT UPDATE
2	MIY FIX
3	VERSION
4	UPDRTE
5	FIX
6	EXP SYSTEM NR.
7	NIGHEST DRT
8	NIGHEST LDEV
9	NIGHEST VDL/M DF VDLS
10	# OF ADO'L DRIVERS
11	CDLO LORO COUNT
12	FILES DUMPEO
13	SERIRL DISC LORD
14	TRPE RECORD SIZE
15	DISC COLD LORD ENTRY
16	MAX INITIRL SEG SIZE
17	SPARE
18	SPARE
19	SPARE
20	DEV CLASS TAB SIZE
21	TERM DESCRIPTOR SIZE
22	DLD VTRB SIZE
23	DLD INFO SIZE
24	CS TABLE SIZE

6.00.00 3- 21

## SYSDUMP/Initial Communication Record (Cont.)

25	SPARE
26	SPRRE
27	SPARE
28	SPRRE
29	SPPRE
30	EDHVERSION BITS HORD 1
31	CONVERSION BITS WORD 2
32	CONVERSION BITS HORD 3
33	CONVERSION BITS HORD 4
34	SPPRE
35	SPARE
36	SPARE
37	SPPRE
38	SPARE
39	SPRRE
40	LOG FILE NUMBER

G.00.00 3- 22

Disc Layout

## Cold Load Information Table Extension

The Cold Load Information Table Extension is a part of the Cold Load Information Table that has no use in booting the system. It exists for different system level processes to hold information that would only be created during a RELORD. A good example of this is the system log file number. This is only created on a RELORD, and changed whenever a log file is full or a boot (other than a RELORD) is performed.

In order to protect the Cold Load Info Table, the extension was created. In this way ND I/Ds should be performed to the Cold Load Information Table during NPE operation. However to process data into the Cold Load Info Extension a process must use the access routine "PROCESS'COLD'LORD'INFO". The exact calling sequence can be found in KERNELD.

The Cold Load Information Extension is 2 sectors long and immediately follows the SYSOUMP/Initial Communication Record starting at sector address #31 on logical device 1.

The assigned entries are as follows:

·	0
RESERVED FOR FUTURE SYSTEM USE	2
1	
	20
SYSTEM LDGGING FILE NUMBER	21
NETWORK MRNRGEMENT LOGGING FILE NUMBER	22
NETWORK MANAGEMENT TRACE FILE NUMBER	23
FULL/PRRTIRL COMMAND DUMP DRTE	24
	25
	26
NOT CURRENTLY RSSIGNED	27
	28
	255

Disc Layout

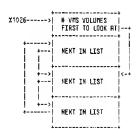
## <u>Virtual Disc Space Management Structures</u>

Disc space for data segments is allocated from reserved regions of system volumes which have been assigned the virtual memory supporting (VMS) attribute. The data structure used for accounting and nanagement of the virtual disc space of the various VMS volumes is the Virtual Disc Space lable (VDSMTAB). This structure consists of a circular list of entries, one for each VMS volume. Each entry contains the information defining the state of the virtual memory region on that volume.

## Virtual Disc Space Management Table

VDSMTRB DST# = 39 (X47) VDSMTRBPTR = Rbsolute(X1D26) = SYSGLOB X26

General Structure



Olsc Layout

VDSMTAB Entry D Format

**VDSMTRB00** TASLE LENGTN # SYSTEM VOLUMES WHICH NAVE VIRTUAL MEMORY VMSVO LUMECHT INDEX OF NEXT ENTAY TO ALLOCRTE FROM VOSMTRB02 STAATENTAY VOSHTRB03 VM PAGE SIZE (512) VMPRGESIZE VOSMTABO4 # SECTORS/VM PAGE (4) SECTORSPERVMPAGE DFFSET FROM ENTRY TO BITHRP (%20) VOSHTABO5 OFFSETTO8M VOSMTABO6 TOTAL N VM PAGES CONFIGURED IN SYSTEM VOSHTABO7 LEAST # DF VM PAGES THAT HAVE EVER BEEN AVAIL. VDSMTAB X10-X17 UNRSSIGNED

Disc Layout

VOSMIAB General Entry Format

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 --|--|--|--|--|--|--|--|--|--|--|--|--INOEX OF NEXT ENTRY IN CIRCULAR LIST Word 0 NEXTINLIST Hord 1 LDEV# LDEV STRATING SECTOR OF DEVICE'S HOSTARTSECTOR VIATUAL MEMORY REGIDN LDSTARTSECTOR Nord 3 Mord 4 # SECTORS IN DEVICE'S TOTAL SECTOR Hord 5 VIRTUAL MEMORY REGION Word 6 # PAGES IN DEVICE'S VIATUAL MEMDAY REGION TOTAL PAGECNT # OF PRGES AVAILABLE IN DEVICE'S VM REGION Word 7 PAGESAVAILA8LE N OF VALID WORDS IN DEVICE'S SIT MAP 8MLENGTN SIZE OF SMRLLEST RECENT MISS Word X11 SMALLESTHISS SMALLEST NUMBER OF PRGES EVER RVRILABLE NOAD X12 UNASSIGNED DEVICE'S VIATUAL MEMORY BIT MAP 11111111111111 111111111111

\*\*\*COMMENT: A bit on in a device's YM8IT MAP
==> Corresponding VM page is free.

G.00.00 3- 25 G.00.00 3- 26

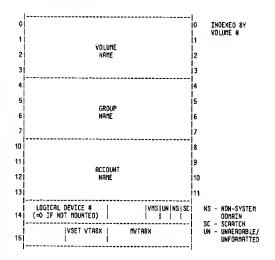
Disc Layout

Volume Table

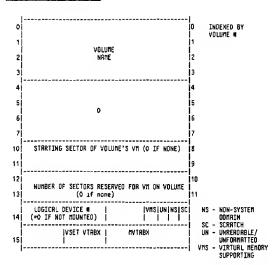
SIA #22=%26 OST #29=%35

Oisc Layout

Typical Private Volume Entry



## Typical System Volume Entry



6.00.00 3- 29 DIRBRSE <----absolute disc addr of base [SYSGLD8+X130 RND X131]

Directory on disc consists of a contiguous area:

DIRBASE -> DIRECTORY BITMAP DIRBASE+3 -> DIRECTORY DATA Entries and Indices

The bitnap defines the available/used sectors in the directory. If the directory is <= 6112 sectors, then the bitnap uill occupy 3 sectors. If the directory size is > 6112 sectors, then the bitnap uill occupy 32 sectors uith DIRMBSC pointing to the 30th sector of the bitnap. A zero bit in the bitnap represents a used sector. Words 0 and 1 of the bitnap are ignored.

Directory entries contain pointers which are sector displacements relative to DIRBASE. Entries and indices are grouped into "blocks".

The capacities for accounts/groups/users/files are dependent on their block sizes.

\* SYSSAIBSIZE SYSRUIBSIZE SYSROIBSIZE SYSGFIBSIZE SYSGFIBSIZE SYSGEBSIZE SYSGEBSIZE SYSGEBSIZE

System acct index block size (3 sectors)
Rcct. user index block size (1-3 sectors)
Rcct. group index block size (1-3 sectors)
Group file index block size (3 sectors)
Group volume set definition ind. blk. size(1 sector)
Rcct. entry block size (3 sectors)
User entry block size (2 sectors)
Group entry block size (2 sectors)
Group entry block size (2 sectors)
Yolume set definition entry block size (1 sector)
Maximum of above. (used to initialize DDS.)

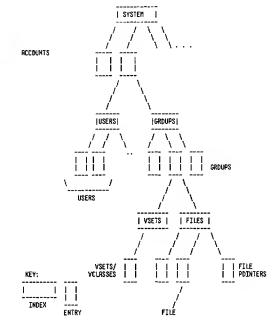
SYSFEBSIZE SYSVSEBSIZE SYSMRXBSIZE

\*These values are used once for the creation of the (root) system, account index or new systems. This root index is always at address DIRBRSE+3.

6.00.00 4- 1

Directory

Dverview of Directory



Dverview of Directory

G.00.00 4- 2

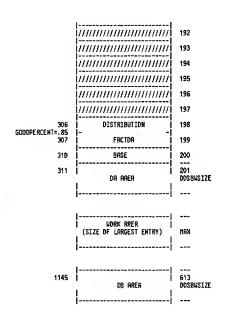
Directory

Directory Data Segment

0		0
:	I SECTOR I Buffer	:
177	128(1D) WDRDS	127
	1 	·
200	ADJUST (D8-DL)	128
201	XTYPE (INPUT PRRM)	129
202	: XHVTABX	130
203	XINDEXP (FINAL INDEX PRT)	131
204	XRNAME (DB REL ADDA)	132
205	XGUNRME (DB REL ADDR)	133
206	XFNAME (DB REL ADDR)	134
207	XRSEC (ACCOUNT SECURITY)	135
21D		136
211	 	137
212	SIRAETURN (FRDM GETSIR)	138
213-240	DIRECTORY POINTER "A"	139-160 \ > SEE Directory
241-266	DIRECTORY POINTER "B"	161-182 / Pointer Area
267	SYS.RCCT.INDEX BLOCK SIZE	183
270	LDEV : DIRECTORY	184
271	PV DIRECTORY SIZE	185
	PRIVRTE VOLUME DIR. SIZE	186
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	187
	///////////////////////////////////////	188
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	189
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	190
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	191

Directory

Directory Data Segment (Cont.)



6.00.00 4-4

### Directory Pointer Rrea [DR or DB] DST=20(10) SIR=8(10)

Ĩ	LDEV   DIRECTORY BASE	139/161 DIRBRSE1'
	ADDRESS OF PAGE IN BUFFER	140/162 DIRBRSE2'
	DIRECTORY PAGE IN BUFFER	141/163 CONTENTS
	DB ADDRESS OF 1ST ELEMENT	142/164 LPNTR
	STARTING ADDRESS OF BUFFER	143/165 IOPNTR
	# VALID PAGES IN BUFFER	144/166 NUMVRLID
-	DI IB	145/167 D=DIRTY FLAG, 8=BRD ELEMENT
-	ELEMENT SIZE	146/16B XSIZE NOTE:
**	# HORDS USED IN BLOCK	147/169 USED ** INCEKES RHD
-	BLOCK SIZE (SECTORS)	148/170 BSIZE
-	BLOCK SIZE (WORDS)	* INDEXES DNLY 149/171 BUSIZE
-	MRX # ELEMENTS/BLOCK	150/172 BFRCTDR
	IIP TY ELEMENT SIZE BLOCK SIZE	151/173 MISCUD
İ	NUMBER OF ELEMENTS	152/174 XCDUNT
į	NUMBER OF ACCESSORS	153/175 PCDUNT
l	ENTRY TOTAL	154/176 ETDTRL
	O P TYLENTRY SIZE BLOCK SIZE	155/177 EMISCUD
İ	FATHER INDEX POINTER	156/17B PINDEXP
į	f	157/179
Ï	T N	158/180 PNAME TY = 0-FILE 1-GROUP
į	E   #	159/181 2-ACCT 3-USER
į		160/182 4-VSD I = 0-ENTRY BLOCK
•		1 = V-ERIKT BLOCK 1-INDEX BLOCK P = PURGE FLAG
		P * PORGC PLNG

6.∞.∞ 4- 5

Directory

## Directory Space Data Segment (DIRSDS)

DST=21 (X25)

SIR=B

DST = 21 ( X25 )

	1 1 1 1 1 1 0 1 2 3 4 5 6 7 8 5 0 1 2 3 4 5	ı
0	Logical device Bit map	
1	baee sector addreee	DS' BRSE
2	Ptr to last avail word in buff	DS'LAST'HORD
3	Ptr to firet word in buffer	DS'FIRST'HORD
4	Size in eectore of directory	DS'DIR'SIZE
5	DIEISIPI	DS'FLAGS
6	First current sector in buff	DS'CUR'SECTOR
7	Disc address of current part	
10	of bit map in the buffer	OS'ADDR
11	Size of buffer in words	DS' SIZE
12	Next requested sector	DS'REQ'SECTOR
13	Last sector in bit map	DS'LAST'SECTOR
14	System saved potr to last	DS'SYS'LRST
15	System eaved potr to firet	DS'SYS'FIRST
16	System saved current sector	DS' SYS' CUR
17	Saved directory size	DS'SYS'SIZE
20	LDEV that last error occurred	DS'ERROR' LDEV
21	Type of error that occurred	DS'ERROR'TYPE

G.00.00 4-6

## Directory

This section of the bit map DST is occupied by up to 3 eectors of bit map. It is swapped in 3 sectors at a time as needed. DSTFIRST WARD is updated to search for space in the bit map. When it reachee DSTMST WARD for the second pase, the next 3 sectors of bit map uill be swapped in.

## Descriptions:

## DS' ADDR

This is the addrsss of the eection of bit map that is currently in the buffers. For example, this addrese uill usually be the same as DS'BRSE. If we need to page in more sectors of bit map than the first three, then thie addrsse uill be eubsequently larger than DS'BRSE.

## DS' BRSE

This is the base address of the directory bit map. If the directory is greater than 6112 sectors, then this address will be 29 sectors less than the address found in the Cold Load Information table on diec.

This ie the current bit map sector number of the first sector in the buffer area. Its value can range from 1 to 30. Thie number minus one added to DS'BRSE will recult in DS'RODR.

## DS'DTR'DTSRRIED

If this bit is on, the directory allocation and deallocation is off and only a WARMSTRRI will turn this bit off. The bit is turned on if an I/D error occurs on a directory bit map sector or if we find data integrity problems with the bit map, i.e. if we attempt to deallocate a eector that is already deallocated.

## Directory

## DS'OIR'SIZE

This ie the eize (sectors) of the directory area. This eize includes only the laet 3 sectors of the bit map. If the directory is greater than 6112 sectors, then this eize doce not include the extra 29 sectors of bit map. It can also be thought of as the number of bits in the bit map.

This bit is eet if the bit map sectors in the buffer have been modified in any way. When more sectors must be brought into the buffers, or if we switch to a different domain (system to PV, PV to system) this bit is interrogated to determine if the sectors presently in the buffers must be first written to disc.

## DS'ERROR' LDEV

The LDEV in which the last directory error occurred.

This word describes the type of directory bit map error that occurred. Ite legal values are:

O - No error
1 - I/D error on a write
2 - I/D error on a read
3 - Rttempting to deallocate space that is already deallocated
4 - Directory space management is already disabled

## DS'ERR'IN'PROGRESS

R directory space management error is currently in progress.

## DS'FIRST'HORD

R DST relative pointer to the word in the bit map buffer that we will interrogate mext when directory space is needed. When the syeten first cones up, this word is always initialized to DSTMERDER-2 (i.e. to point to the first word in the bit map). On subsequent bit map sector reads, it is set to DSTMERDER since subsequent eectore will not have the 2 word overhead that exists in the first sector of the bit map.

## DS'FLAGS

This word contains numerous flags. See individual descriptions.

## DS'LAST'SECTOR

This is the total number of active bit map sectors. This number uill range from 1 to 32.

This is the current number of bit map word in the buffer. It can range from 1 to X577 + DS'HERDER. If there exists 3 full sectors in the buffer, then it will have the value X600 + OS'HERDER - 1 or X621. It is compared to DS'FIRST'MDRD to determine if we have hit the end of the current buffer

DS'PERM'DISABLE

If this bit is set, then directory allocation/deallocating is permanently disabled. This bit should not be set.

DS'REQ'SECTOR

This is the next sector to begin reading in up to 3 bit map sectors. It is updated by 2 or 3 and the read procedure will bring in up to 3 sectors starting from this sector. If this sector is set to be greater than OS'LRSI'SECTOR, then it is reset to 1. Rfter the sectors are read in, DS'CUR'SECTOR is set the OS'REQ'SECTOR.

This is the size in words of the bit map buffer area. It is always a multiple of a sector (128 words). It will usually have the value of %600. legal values are %200, %400 and %600.

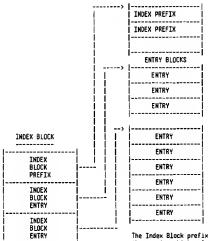
DS'SYS'LAST, DS'SYS'FIRST, DS'SYS'CUR & OS'SYS'SIZE

The values of DS'LAST'MORO, DS'FIRST'MORO, OS'CUR'SECTOR and DS'SIZE will be stored in these locations when the directory space management switches from the system directory to a private volume directory. Rnd, of course, when DSN switches back to system domain, the above mentioned values are reinitialized with these values.

Directory

### Directory Structure

INDEX BLDCK



The Index Block prefix points back to the previous higher level. The Index Block entries point to the entry blocks.

Directory

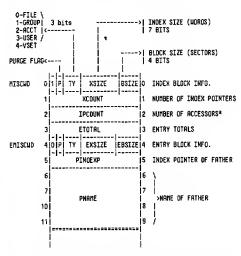
## Directory Definitions

- smallest allocatable record ("phys.recd")-currently sector.
- integral# of pages; contains contiguous indices or entries.
- pointer to entry block, containing name of 1st entry.
- information-containing "object" may contain pointer to an index block.
- 15-bit positive relative page number (relative to directory hape) >PAGE >BLOCK >INDEX >ENTRY

base).

>DDS - directory data segment. >ELEMENT - a generic name for index or entry.

Index Block Prefix (10 Words)

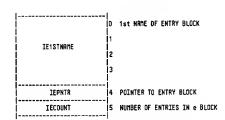


\*The count is incremented by each access that uses and relies upon a pointer to the index block, i.e., lt is guaranteed not to be purged while the count is not = 0.

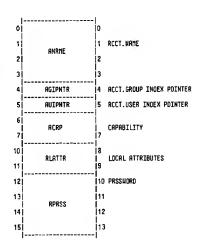
Directory

Index Entry (6 Words)

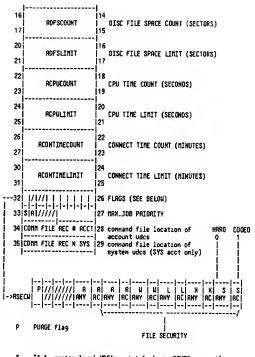
G.00.00 4- 10



Account Entry (236 Words)



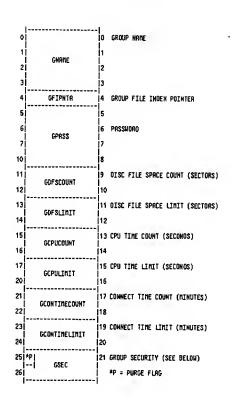
## Account Entry (Cont.)



S If 1, system level UDC's exist (only in "SYS" account) R If 1, account level UDC's exist for account

6.00.00 4- 13

## Group Entry (Z51 Words)



6.00.00 4- 14

Orrectory

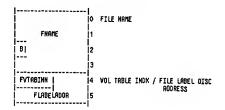
## Group Entry (Cont.)

		1	
27	GCRPRBILITY	23	GROUP CAPABILITY
30	GLINKAGE	24	GROUP DIA. BASE LINKAGE
31	GVSDIPNTR	25	GROUP VOL SET DEFN INDN
32	GHVSNAME	26	NOME VOL SET NAME
33	-	27	
34	GHVSANAME	28	(Definition's acct mame)
35	-	29	
36		30	
37	Old Incollege	31	
40	- GHVSCHANE -	32	(Definition's group mame)
41	-	33	
42		34	
43	-	35	
44	- GHVSVSNAME -	36	(Definition's vol set name)
45	-	37	
46	GSAVEFIPNTA	38	SAVE CELL FOR GFIPHTR
47	GHOUNTREFCNTA	39	GROUP BIND COUNTER
50	0	40	GSPARE
ı			

Directory

## Group Entry (Cont.)

## File Entry (File Pointer)(6 Words)



B - Bad file label (0:1) = 0 - not defective = 1 - defective

## User Entry (19 Words)

USER HAME 4 CAPABILITY UCRP si 6 LOCAL ATTRIBUTES ULRITR 10 İ 8 PASSUDRO UPASS 110 12 HOME GROUP (MAY BE NULL) 15 16 17 UHGROUP 17 | 15 | LOG CMT (# OF USERS LOGGED OM)
20 | ULOGCOUNT | 16 INIT TO 1 FOR MANAGER. SYS SO
UMAXJOBU 21 PIUU 0 | JORPAT | 17 MAX. JOB PRI 1/47 = UNDE FLAG 22|CONT FILE REC # | (command file loc of user udcs) 118

User Attributes/Capability

SAVE FILES ---FILE-ACCESS ATTRIBUTES 4 -----ACCOUNT LIBRN
-----GROUP LIBRN
-----DIRGNOSTICIAN
---SYSTEM SUPVSR
| TREATE VOLS USER / batch access interactive access privileged mode ACCESS privilege-10 SEMERAL | ultiple PINS RESOURCES extra data segment \ process handling ----

G.00.00 4- 17

G. 20.00 4-18

## Directory

## Volume Set Definition Entry

VOLUME SET CYSHAME TY = 0 4 TY | R12 MVT98X GVSLINKAGE 15 GVSINFO SIVOL COUNTIA 71 VERSK | 6 MEMBER VOLUME | 7 MAME(1ST ENTRY | 8 IS MASTER | 9 VOLUME) VOLUME ENTRY O (6 HORDS) GAZAOTINE 111 120 141 HI10 GVSVOLFLAGS \ 13| PSEUDO SUBTYPE 11 GVSVOLINFO 12 VOLUME ENTRIES 1 - 7 47 \ **5**7 42 60 49 50 MEM, VOL. 62 GVSVOLUME | |51 63 GVSVOLFLAGS (MEMBER VOLUME FLAGS) 52 (MEMBER VOLUME INFO) 53 GVSVOLINFO 65 İ54 GYSDREFONT (DEFN. REF. CHTR.) SS SPRRE

VIABX: VOLUME TABLE INDEX

## GVSLINKAGE

0 1 2 3 4 5 6 7 8 9 10 19 12 13 14 15 TIA NOT

0 = Volume Set Definition 1 = Volume Set Class 8 = RELOCATING FEAG

MYTABX - No

## GVSINFO

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 NOT USEO VSMASK

VOLCHT - Number of members in set VSTASK - Bit mask of volume member usage Order is from right to left i.e., bit 15 is 1st member, bit 14 iv 2nd member ...

## GVSVOLFLAGS

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 HOT USED

N - Nember Mounted Flag
0 = not mounted
1 = mounted

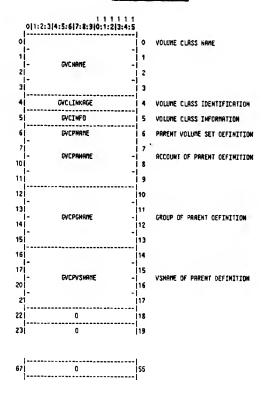
## GVSVOLINFO

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 DISC PSEUDO SUBTYPE

DISC PSEUDO-SUBTYPE = (Actual type \*16) + actual subtype. VTABX - Volume Table Index

G.00.00 4- 20

## Volume Set Class Entry



G.00.00 4- 21

GVCLINKAGE

0 1	\$													
11	1////	////	////	11111	////	////	////	////	///	////	1///	////	1///	///

1 - TYPE 1 = Volume Set Definition 0 = Volume Set Clase

GVCINFO

0	1	S	3	4	5	6	7	8	9	10	11	12	13	14	15
	VOL	CNT			NC US	T ED					VC	MASK			

VOLCMT - Mumber of members in set
VCMRSK - Bit mask of volume member usage (VOLUME CLRSS MRSK)
Order is from right to left
i.e. bit 15 is 1st member, bit 14 ie 2nd member ...

Volume Mask Format

- USEO IM MYTAB, PVUSER, FILE CONTROL BLOCK (FCB), VOLUME SET/CLRSS DEFINITION, VOLUME SET VTAB. - 8-BIT TASK.

47	V6	l VS	V4	V3	V2	٧1	1 40	1	
	~	~	~	~	~~	~			
		ļ		1	ļ		1	VOLUME	O (MASTER)
İ	İ	İ	ļ	İ	į			VOLUME	1
İ	į	İ	į	İ	<u>-</u>			VOLUME	2
1	İ	-	-					VO LUME	_
-	-	1						VOLUME	
-	1							VOLUME	
-								VOLUME	·
MOT	HOUNT	EA N	שחש.	afapc				VOLUME	7 NEMBER

G.00.00 4- 22

## CHAPTER 5 LOCK RESOURCES

## SIR# Rilocation DST 253

Sir's Ordere	d by Sir Number	
SIR #	RANK	SIR MAME
1	10	LORO PROCESS
	335	CACHE CONTROL
3	91	100
2 3 4	92	000
5	50	PROCESS TREE STRUCTURE
6	60	SCHEDULING QUEUE
5 5 7 8	70	CST ENTRIES
8	80	SYSTEM DIRECTORY
9	90	LPDT
10	B5	LDT
11	110	STORRGE IN OVERLAY AREA
13	130	JPCNT
14	140	JCUT
15	27	JNAT
16	5	FMRVT LOADER SEGNENT TRBLE
17	22	ADD FOHDEK SERVEMI JUDITE
18	180	SP00L
19 20	190 200	MESSAGE CATRLOGUE
20 21	210	RIT
21	220	VOLUME TABLE
23	230	WELCOME MESSAGE SIR
24	240	ASSOCIATION TABLE
25	250	CS RLLOCATE
26	260	LOGGING BUFFER
27	83	PV MVTAB
28	280	MERSSIR
29	290	PV USER TABLE
30	300	IMAGE
31	310	KSAM
32	320	USER LOGGING
33	330	DEBUG BREAKPOINT TRBLE
34	340	PCB
35	350	SUB-QUEUE MAPPING TRBLE
36	360	CILOG
37	25	FILE INTEGRITY
3B	380	RIN
39	390	TAPE LABELS
40	87	DEVICE CLASS TABLE
41	400	Reserved
42	401	Cold Load SIR
43		1st J08
44		2nd JOB
	•	•
		•

G.00.00 5- 1

## Sir's Ordered by Ranking

RANK	SIR N	SIR NAME
5	76 -	FMRVT
10	1	LORO PROCESS
žž	17	LORDER SEGMENT TABLE
25	37	FILE INTEGRITY
	15	JHAT
27		PROCESS TREE STRUCTURE
50	2	
60	5 6 7	SCHEDULING QUEUE
70	7	CST ENTRIES
80	8	SYSTEM DIRECTORY
B3	27	PV MVTAB.
B5	10	LOT
B7	40	DEVICE CLASS TRBLE
90	9	LPDT
91	3	IDD
92	4	000
110	11	STORRGE IN OVERLAY AREA
130	13	JPCNT
140	14	JCUT
180	18	V00
190	19	SPOOK
200	20	MESSAGE CATALOG
210	21	RIT
220	22	VOLUME TABLE
230	23	HELCOME MESSAGE
230 240	24	ASSOCIATION TABLE
		CS ALLOCATE
250	25	LOGGING BUFFER
260	26	MEASSIR
280	2B	PV USER TABLE
290	29	
300	30	INAGE
310	31	KSAM
320	32	USER LOGGING
330	33	DEBUG BREAKPOINT TABLE
335	2	CACHE CONTROL
340	34	PCB
350	35	SUB-QUEUE MAPPING TABLE
360	36	CILOG
380	38	RIN
390	39	TAPE LABELS
400	41	Reserved

G.00.00 5- 2

Lock Resources

## SIR Table Information

The system internal resource table is located in non-linked memory (resident table). The SIR table is used to protect critical system elements against access by more than one process, i.e., it provides a "lock out" mechanism. Each critical system resource (usually a table) is assigned a specific SIR number. Procedures are provided within MPE to lock (GETSIR) and unlock (REISIR) the SIR. Processes attempting to obtain a SIR that is not available are impeded by the system. The SIR table entries from the head of a linked list in this case. If more than one process becomes impeded, word 15 of the PDB entry is used to add the "new" process to the growing list. The method of unimpeding the process depends on the SIR type.

A SIR does not respect process priority and operates in a FIFO manner. As processes become impeded on behalf of a SIR the new entries are entered at the tail of the impeded list. When the current holder of the SIR releases it, on the first process in the list (pointed at by the head pointer) is unimpeded. The linked list head and all pointers are then updated and the newly unimpeded process will obtain the SIR.

Lock Resources

## SIR Entry Formats

## 0 1 2 3 4 5 6 7 B 9 10 11 12 13 14 15

<b>          </b>	O free
0	1 (not locked)
0	2
0	3
	I I
PCB index of holder	0 SIR locked
0	1 (no impeded processes
0	2
0	3
	1
PCB index of holder	0 SIR locked
SIR QUEUE LENGTH	1 (inpeded processes)
HERO OF IMPEDED LIST(PCB relative)	2
TRIL OF IMPEDEO LIST(PCO relative)	3
	1

P = PIN# PIN = PCB table entry number SIR QUEUE LENGIH- number of processes queued for this SIR

The SIR table is indexed by SIRB, with each SIRM corresponding to a unique, pre-assigned system internal resource. Entry #0 is not used. Inpeded lists are established by using the SIR table entry (2) as the head of the list and PCB(15) for elements. PINs are always used as pointers, with 0 indicating end of list.

### CHAPTER 6 FILE SYSTEM

This chapter describes the MPE V file system. The second section describes the basic concepts. The third section describes the table structures used.

#### File System Overview

The object overview

I/O to files is done by reference to file numbers, which are assigned by calling the FOPEN intrinsic. This establishes an initial "point of attachnent", which may be described as a connection between a program (i.e., process) and that particular point in a particular file at which the next FRERO or FURITE would cause data to be transferred. A point of attachnent is described by a control block, of which there are several different kinds described later). Control blocks may exist in the process's own stack or in an extra data segment assigned by the file system. In order to find control blocks may exist in the process's own stack or in an extra data segment assigned by the file system. In order to find control blocks by a vector, which consists of two words with the first word containing a segment number and the second word containing a word offset into the control block within that segment. The entire assemblage, consisting of eight overhead words, the vector table, and all of the control blocks to which it points, comprises the entire segment; if in a stack, it occupies part of the PXFILE part of the PCBX.

The point of attachment is described by a "physical access control block", or PRCB, which will exist as a result of an FOPEN to any file (except \$NULL). Any required I/O buffers are associated with the PRCB; refer to Section 2.1.

All FOPENs specifying "nulti-access" for all processes running under a single plob use a single PRCB for references to a nulti-access file. Although all these are attached to a single point in the file, the type of attachment (i.e., ROPTIONS) may be different. So, each FOPEN specifying a nulti-access file establishes a "logical access control block", or LHCB, which contains the point-of-attachment local values. The use of a single buffer (i.e., PRCB) ensures that references by various processes or against various FOPENs within one process are dealt with in strict sequential order. Note that references to a file by other jobs, or by other processes not specifying multi-access, will be through other PRCBs, whose buffers will be read or written at the pleasure of the file system; in order to ensure any sort of and FUNLOCK the file. \$STOIN, \$STOLIST, and spoolfiles are opened multi-access automatically.

In the case of disc files, there is another kind of control block: the file control block (FCB). It contains copies of information read from the file label, such as the end-of-file pointer, the extent map, and the record and block structure. The EOF pointer is updated in the FCB as the file is uritten, and all changes made to the FCB are posted to the file label when the file is closed. An FCB is shared by all jobs in the system which reference the file

G.00.00

File System

## Table Formats

This section gives a detailed discussion of the nain tables constructed and used by the file system. The location and overall structure of each table is given, in addition to the table format and a discussion of each field in the table. Table indices at the right of the table are in octal. Index names apply to the entire word; if in parentheses, the names are defined in the file system listing but not explicitly used there.

## File System Section of PCBX (PXFILE)

The PXFILE area is a subsection of the PCBX. It is a contiguous, expandable and contractible block of storage that is nanaged by the file system primarily for its own use. Other subsystems, nanely (S and DS, also nake use of the PXFILE section. In doing so they must conform to the conventions of the file

The overall structure of the PXFILE area is:

	-
OVERHEAD	(FIXED)
CONTROL BLOCX TRBLE	(VARIABLE)
RVAILABLE	(VARIRBLE)
ACTIVE FILE TABLE	(VARIABLE) DL-5

File System

The file number assigned by an FOPEN is an index into the Available File Table (AFT), a table of six-word entries which is at the end of the PXFILE part of the PCBX. Two double words are vectors to the PACB and (if it ex-

RFT entries can also reside in a global RFT extra data segment. If the file was opened Global RFT (specified in the RGPTIONS) and the program is privileged, then the RFT is placed into this global RFT OST. Rny accesses to the file are identical to local RFT's. Rll accesses to the file opened global must be done from privilege mode code. The file system intrinsics distinguish this file by a negative file number. Rgain, these files are identical in every other way except for where the RFT entry resides.

Because control blocks are shared among processes, it is necessary to have a schene for coordinating access to then. A control block is "locked" by a process which requires exclusive access to it for a time. Other processes which attempt to lock the block will find it already locked, and will be impeded and queued. It may also be necessary to lock an entire control block table so that a process can create or destroy a control block in it, or lock or unlock an existing control block in the table.

Rnother table used by FOPEN is the File Multi-Access Vector Table (FMRVT). This table exists in a system extra data segment and is used by all jobs and processes in the system. When a file is being FOPENed with multi-access specified, the FMRVT is searched; if the file is already open, the FMRVT gives the PACB vector for the prior reference for each job.

### Buffers

A bit in AOPTIONS specifies, when a file is opened, whether access is to be buffered or unbuffered. If unbuffered, data is transferred directly between the I/O device and the user's buffer (usually in his stack), which will be frozen in nenory for the duration of the transfer. If buffered, the data is noved between the user's buffer and a file system buffer to which the I/O is actually done.

Buffers are associated with the PACB, attached to it as an appendage.

G.00.00

File System

## Overhead

The part labeled Overhead contains information that pertains to the entire section. It is addressed via the pointer at OL-3.

	0 1 7 8 15		
•	PXFILE SIZE IN WORDS	0	PXF\$IZE
	LAST OOPEN ERROR NO.   LAST COPEN ERROR NO.	1	
	N	2	
	LAST DE RFT	3	
	SLRVE AFT NUMBER	4	
	LAST KOPEN ERROR NUMBER   LAST FOPEN ERROR NUMBER	5	
	AFT SIZE IN HORDS	6	PXRFTSIZE
	00 70005 5715 7850	7	(PXCTRINFO)
	CS TRACE FILE INFO	8	(PACIKINFO)
	LAST RESPONDING NO-HAIT I/O AFT ENTRY NUMBER	9	PXFLEFT0FF
	1ST USER (MOBUF) CONTROL BLOCK TABLE DST MUNGER	10	PXFCBT1
	2NO USER (NOBUF) CONTROL BLOCK TRBLE OST NUMBER	11	(PXFCBT2)
	3RO USER (NOBUF) CONTROL BLOCK TABLE OST NUMBER	12	(PXFCBT3)
	4TH USER (NOBUF) CONTROL BLOCK TABLE OST NUMBER	13	(PXFCBT4)
	STH USER (NOBUF) CONTROL BLOCK TABLE OST NUMBER	14	(PXFCBT5)
	6TH USER (NOBUF) CONTROL BLOCX TRBLE OST NUMBER	15	(PXFCBT6)
	7TH USER (NOBUF) CONTROL BLOCX TABLE OST NUMBER	16	(PXFCBT7)
	8TH USER (NOBUF) CONTROL BLOCX TABLE OST NUMBER	17	(PXFCBT8)

Partial word field identifiers are:

PXFDOPEN	= PXFILE(1).(0:8)#,	last DOPEN error code
PXFCDPEN	= PXFILE(1).(8:8)#,	last COPEN error code
PXFNOCB	= PXFILE(2).(0:1)#,	no CB's in PXFILE CBT?
PXFKOPEN	= PXFILE(5).(0:8)#,	last KOPEN error code
PXFFOPEN	= PXFILE(5).(8:8)#,	last FOPEN error code

Discueelon:

PXENDER

This is the size (in words) of the Active File Table (AFT). The size ie in words to simplify calculating the size of the available block. PXFAFTS1ZE

These are the DST numbers of the user (NDBUF) control block tables. A DST number of 0 indicates that no data segment is PXFCBT1-8

PXFCOPEN This contains the last CDPEN error number. Not used by the file system.

PXECTAINFO This contains information pertinent to the CS trace file. Not used by the file system.

This contains the last DDPEN error number. Not used by the file system. PXFDOPEN

PXFDSINFD Reserved for DS. Not used by the File system.

This contains the last FOPEN error number. If it is zero then the last FOPEN successfully completed; otherwise the last FOPEN was unsuccessful and the number is the file sys-PXFFDPEN

DXFKDDFN

This contains the last KDPEN error number. XSRM is partly embedded in the file system, and an FDPEN failure on a KSRM file can be caused by a failure to open either the key file or the data file. This error number is used in conjunction with DWFFDPEN to determine which file caused the XSRM open failure. This error number is not used by the file system.

This is the RFT entry number of the last file/line that completed a nowait 1/0; if zero then no nowait 1/0 has been completed. This cell is maintained solely by and for the IOMATI intrinsic. PXFLEFTDFF

This bit signifies that control blocks are not to be created in the PXFILE control block table. This bit is set by the NDCB parameter to the CRENTE intrinsic or the :RNN command. This feature permits the user to have as nuch stack space as possible; otherwise the file system uil take several hundred words of stack for the PXFILE control block table.

This is the size (in words) of the complete PKFILE area. It is the sum of the overhead block, the control block table, the active file table and the available block. PXFSIZE

PXFILE Control Block Table (PXFCBT)

Addressing within a PXFILE control block table is somewhat more complicated than addressing an extra data segment CBT since the table does not begin at DB+O. Rs a result all pointers within the table are table relative; the starting address of the table must be added to a pointer to generate a final DB-relative address. This addressing convention is consistently applied to all control block tables.

When the control block table is expanded, space is taken from the RVRILABLE area. If no space is available then the PWFILE area is expanded and the acquired space is added to the RVRILABLE area.

The part labeled Available is used to provide space when the Control Block Table or the Active File Table is expanded. These two tables grow towards each other and when more space is needed it is simply taken from the Available Block.

When the Available area is exhausted, the PXFILE area is expanded, the AFT is relocated and the new space is added to the Available Block.

Currently the PXFILE area is only expanded; it is never contracted.

File System

Active File Table (AFT)

The part labeled flotive File Table contains information used by the file system (or CS, DS, etc.) to grossly characterize the file access and, most importantly, to give the location of the control blocks.

The overall structure of the AFT is:

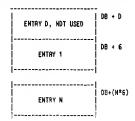
(FIXED, 6 WDROS) ENTRY N DL-9 (FIXED) DL-5 ENTRY 1

where N = PXFRFTSIZE/6.

The length of the RFT ie specified by PXFAFTSIZE. Unused entries are all zeros. When the table is full it is expanded by taking space from the Available block.

The RFT is negatively indexed by file number: the entry at DL-9 corresponds to file number 1, the entry at DL-15 corresponds to file number 2, etc.

The structure of the global AFT DST, described in Section 2 is as follows:



File System

The etructure of a file system AFT entry is:

0 1 2 3 4 5 15		
ENTAY TYPE   N	0	
PNYSICAL ACB DST NUMBEA	1	AFTPACBOST
PHYSICAL RCB ENTRY ADDRESS	2	AFTPACBENTRY
LOGICAL ACB DST MUMBER	3	RFTLACEDST
LDGICAL ACB ENTRY ADDRESS	4	AFTLACBENTAY
ND-URIT I/D IOQX	5	AFT10QX

G.∞.∞ 6- 6

The entry Format depends on the entry type; the File system uses entry type

The Following partial word Field identifiers are used:

shull file

Discussion: AFTINDX

**AFTLACBENTRY** 

This is the IDO index of the pending nowait I/D (if any). This is applicable if the file was opened with the NDWHI option specified. Also, CS and DS have the same capability and use this cell in a consistent namner. This is because the IDWHII intrinsic services the file system as well as CS and DS, and is the principal user of this cell. If the IDOW is negative, then one of two possibilities exist. If the file is a message file, then file IDOW is the accessor's reply port. If the file is a standard NPE file, then a read was done to a nonexistent extent and this is simply a stub inserted by the file system.

This is the DST that the Logical RCB (LACB) if it exists. This is applicable if the file was opened with the multi-access option specified. **AFTLACBOST** 

This is the word offset into the control block table of the LACB vector table entry, applicable if the file was opened with the multi-access option specified.

RETNULL This bit signifies that the file is \$WULL and that there are no control blocks. File System

**RETPACEDST** 

This is the DST that contains the Physical ACB (PACB). A PACS exists for all files except \$NULL.

RETPROBENTRY

This is the word offset into the control block table of the PRCB vector table entry. This will be nonzero for all files except \$NULL.

RFTTYPE

This is the AFT entry type number. At present the following entry types are defined:

O - file system
1 - remote file
2 - OS (nowait I/D disallowed)
3 - OS (nowait I/D allowed)
4 - CS
5 - CS

6 - KSAM 8 - Message File

Remote file RFT entry:

```
LINE NUMBER
      REMOTE FILE NUMBER
   PENDING FOLOSE DISPOSITION FROM FOPEN
                              3
          UNUSED
           IOQX
```

RFT 0
FSTYPE - This value will be 1 for renote files.
RR - Set if the file was opened multi-access.
RFT 1 - Local line number of renote file.
RFT 2 - File number of the renote file.
RFT 3 - Pending disposition of the file. Set when file was FOPEN'd and will possibly be used as the FCLOSE disposition.
RFT 5 - No wait I/O Queue Index.

G.00.00 6- 9

File System

CS Line entry:

```
LOGICAL DEVICE NUMBER
     VECTOR TO MULTIPLE TOQ INDICES
                                 2
                   UNUSEO
                                 3
TR | I| R| DIAL|
          MISC'DST
         IDQX ( CID only )
```

RFT 0
FTYPE - This value will be 4 or 5. R 5 signifies that the line has an autodialer attached.
R - The line has been opened with no waiting on I/O requests.
ID - Line is a multipoint control or 3270 station.
B - Line was opened with buffering.
RFT 1 - Logical device number of the line.
RFT 2 - Vector to Multiple IOQ indices.

TR - Bit O on signifies tracing enabled. Bit 1 on signifies trace all.

I - On if line is currently connected.

R - Signifies that this CS device is an SCCP device.

DIRL - O = Dial on write, answer on read.

1 = Rhsuer on write, diel on read.

2 = Rhusys dial.

14 - DST number of the line's misc data segment.

15 - If <> O, then it is the system DB address of a single request IOQ entry. IOUNIT uses this word to pass the IOQ index of the completed request for this RFT to CSIONART.

File System

OS AFT entry:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15					
FSTYPE   C  H  P  R  OS ERROR NUMBER	0				
ORTH SEGMENT NUMBER					
OSOCB INDEX   UNUSED	2				
LDEV NUMBER					
PREVIOUS AFT POINTER					
IDQX					
	FSTYPE C I II PI RI OS ERROR NUMBER  ORTH SECHENT NUMBER  OSOCB INDEX   UNUSED  LDEV NUMBER  PREVIOUS AFT POINTER				

FST 0

FSTYPE - This field will have the value 2 or 3.

C - On if DSSPEN called by CXDSLINE or REMOTE MELLO.

M - On if Master PIDP RFT.

P - On if PIDP related.

R - Dn if remote main process.

RFT 1 - DS data segment table pointer.

RFT 2 - DSDSCB Index - DS data segment control block index.

RFT 3 - Logical device number.

RFT 4 - Preceding DS open RFT Pointer.

RFT 5 - IOOX - Same as described above.

6.00.00 6- 10

File System

## File Control Block Table (EBTAB)

A file control block table can be located in two places: (a) as a subpart of the PXFILE area, as discussed in Section 3.1.2; or (b) in a data segment. Although purting control block tables in PXFILE has the advantage of providing rapid access, it detracts from the space for the user's stack; so the larger control blocks (or optionally, all control blocks) are put into extra data segments. On the other hand, referencing extra data segment control block tables are of three kinds: expandable, nonexpandable, and shared FCB. Monexpandable (BT's are used for a single PXGC with buffers, i.e., where the control block is large or where the control block can't be local to a single process (for multi-access). Expandable (or NBBIF) CBT's are used for a single process (BTB's, PRCB's with no buffers, and FCB's which are local to a single process. It list of the expandable CBT's associated with small control block is needed, these CBT's are checked in order to see if one of then has room. Shared FCB CBT's are checked in order to see if one of then has room. Shared FCB CBT's are checked in order to see if one of then has room. Shared FCB CBT's are similar to expandable CBT's except that they belong to the system rather than to a single process; the system keeps a list of DST's which it has assigned for this purpose.

The overall structure of a control block table is:

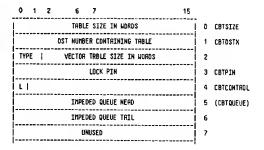
OVERHEAD	(FIXEO, 8 HDRDS)
VECTOR TABLE	(VARIABLE)
CONTROL BLOCK RREA	(VARIABLE)

G.00.00 6- 12

6.00.00 6- 11

#### Overhead

The part labeled Overhead contains information pertaining to the entire table.



Other identifiers used:

CBTTYPE = CBTAB(2).(0:2) Control block table type
CBTVTSIZE = CBTAB(2).(2:14) Vector table size
CBTLOCKBIT= CBTCONTROL.(0:1) Lock bit

COTOSTX

This is the DST number of the data segment that contains the control block table. If the table is contained in a stack, i.e. in the PXFILE area, then this is the DST number of the stack and not 0.

**CBTLOCKBIT** 

If the entire control block table is locked, then this bit is set. No locking count is kept since control blocks are locked only once from FCREFIECB and FDELFEECB when control blocks are added to and deleted from the table. The procedure LOCK/FCB does not lock the control block because it runs PSEUDODISABLED during the critical times.

CATOUFUE

This is the inpeded queue for the table and has the same format as the impeded queue for a control block in the table. There is no second impeded queue because that facility is used exclusively for BRERK requests against the PRCB for \$STDIM/\$STDLIST.

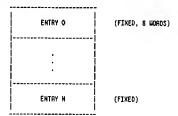
6.00.00 6- 13

File System File System

## **Vector Table**

The part labeled Vector Table contains information used to locate and lock or unlock control blocks in the control block table.

The overall structure of the vector table is:



where H = (CBTVTSIZE/B)-1.

An unused vector table entry will have zeros in all the words of the entry. A used vector table entry will have a nonzero value in the first word of the entry (the control block address is necessarily nonzero).

The general structure of a vector table entry is:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15		
	١٠	VT'RDA
L  B  COUNT   UNUSEO	1	VT'CONTADL
LDCK PIN	2	VT'PIN
NIGH PRIDRITY NERD PIN	3	YT'QNERD
NIGH PRIDRITY TRIL PIN	4	VT'QTRIL
LOW PRIORITY HEAD PIN	5	YT' SRVEDHERO
LOW PRIGALTY TAIL PIN	6	VT'SAVEDTAIL
UNUSED	7	
	[	

file System

This is the PIM number of the process that has the control block locked. CBTPIN

This is the size in words of the table. It is in-itialized when the table is created and changed when the table is expanded. Rt present a table is never contrac-ted, even though this is possible. CBTS17F

CBTTYPE This field is the type of the control block table. Possible values are:

O - stack [PXFILE]
1 - NDBUF (expandable)
2 - System shared FCB
3 - Buffered (Contains a single PRCB)

This is the size, in words, of the vector table area in the control block table. It does not reflect the number of entries used or unused. CBTV TSIZE

 $\mbox{\sc MOTE:} \ \mbox{\sc All PIN's are kept as the word offset into the PCB table and as the actual PIN number.}$ 

G.00.00

The following partial word identifiers are used:

VT'LOCX'BIT = VT'CONTADL.(0:1)
VT'BREAK'BIT = VT'CONTADL.(1:1)
VT'COUNT = VT'CONTADL.(2:6)

Oiscussion:

Control block address is the table relative address of the control block associated with the vector table entry. It is a word displacement from the beginning of the control VT'ROR

This bit signifies that we are in the middle of break mode. This is used for the PRCB of \$STDIM/\$STDLIST from a terminal session only. VI'BRERK'BIT

VT'LOCX'BIT

This bit is set whenever the control block is locked. VI'COUNT This is the count of the number of times that the control block has been locked by the process identified in VT'PIN. If it is zero, then the control block is not locked.

VT'PIN

Contains the PIN of the process which has exclusive access to the control block. Other processes attempting to access the block will be impeded and

The high priority inpeded queue is a double word of PINs that are the head and tail of the impeded queue of processes waiting for access to the control block. Processes are inpeded and unimpeded by the file system using the normal nechanisms available under MPE. VT'OUEUE

VT'SRVEDDUEUE

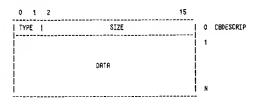
The low priority inpeded queue is a double word of PINs and has the same format as VTQUEUE. The only time this word is used is when the control block is in BRERK node, which can only happen to an RCB corresponding to \$STDIM/\$STDLIST. It is used to save the current VT'OUEUE when the control block goes into BRERK wode and to restore VT'OUEUE when the control block goes back into non-BRERK mode.

NOTE: PI1 PIN's are stored as offsets within the PCB table and not as actual PIN numbers.

## Control Block Area

The part labeled CONTROL BLOCK RREA contains the control blocks used by the file system.

To facilitate storage management, all control blocks have the same overall structure:



where N = Size-1.

Partial word field identifiers are:

= CB.(0:2)#, = CB.(2:14)#; control block type number. CBTYPE

Oiscussion:

CBDESCRIP This is the first word of a control block; the format is common for all control blocks.

CBSIZE This is the size (in words) of the control block. The size

includes the descriptor word.

This is the type number of the control block. There are four types of control blocks: CBTYPE

0 - Garbage 1 - FCB 2 - PACB 3 - LACB

When a control block table is created the initial control block area is completely allocated to a single control block of type garbage. When space is requested for a new control block the control block area is scanned (using a first fit algorithm) for a garbage control block that is as large as the size requested. The space for the new control block is taken from this garbage control block and the space remaining becomes the new garbage control block size.

G.00.00 6- 17

File System

When space is returned it becomes a new garbage control block. To reduce fragmentation the new garbage control block is combined with either of the two neighboring control blocks if they are of type garbage.

If space is requested and no garbage control block is large enough to contain the new control block then the control block area and control block table are expanded by a sufficient amount. If expansion is not possible, some other control block table must be used.

## Access Control Block (ACB)

Virtually every file system intrinsic constructs an ACB as its first action. When using the multi-access option, each accessor shares a single PACB. However each accessor is permitted to view the shared file in a slightly different manner than the other accessors. For example, one accessor may access the file in a read-only mode while the other accessors may access the file in a read-urite mode. To do this, each accessor must, during his access, have a slightly different ACB. slightly different ACB.

The PACB holds information that is global to all accessors of the file. The LACB holds information that is local to each accessor of the file. At the beginning of a particular access, an ACB is constructed by calling LOC'ACB, which copies information from both the LACB and the PACB. At the end of the access, the ACB is released by calling UNLOC'ACB, this updates the PACB ACB LACB from the RCB since some of the fields may have been modified due to the access. This scheme nearly eliminates EXCMANGEDB's to access the various data segments.

G.00.00 6- 18

File System

## Logical Access Control Block (LACB)

All LACBs have the same structure:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

3   COMPLETE LACE SIZE	0				
FILE NUMBER	1				
FILE NAME - 1ST CNAR.   FILE HAME - 2NO CHAR.	2				
FILE NAME - 3RD CNAR.   FILE NAME - 4TH CHAR.	3				
FILE NAME - 5TH CHAR.   FILE NAME - 6TH CHAR.	4				
FILE NAME - 7TN CHAR.   FILE HAME - BTH CHAR.	5				
FOPTIONS	6				
AOPTIOHS	7				
RECORD SIZE IN-BYTES	10				
BLOCK SIZE IH WORDS	11				
SPARE	12				
CARRIAGE CONTROL CODE	13				
EDF PG  LN  ST  FK  TC  TB  8B  CAR[DB   EOF T   EOF M	14				
C!   TEL IC! Q     TERMINAL STOP CHARACTER	15				
ERROR CODE					
LAST I/O TRANSMISSION LOG	17				

Partial word field identifiers are:

LACBSIZE = LACB.(2:14)#, size in words
LACBSTOPCHAR = LACB(2).(0:B)#, terminal stop character

6.00.00 6-19

Discussion:

See ACBROPTIONS. LACABORTIONS LACBBSIZE See ACBBSIZE.

File System

LACOSTATE

See ACBCTL. LACECTL

See ACBERROR. LACBERROR

LACBENUM See ACBFHUM.

LACBFORTIONS See ACBFOPTIONS.

LACBMODE See ACBMODE. LACBNAME1-8

See ACBNAME. LACBPACE

This is the DST and vector table entry for the Physical ACB (PACB) for the file. Physical file.

LACBRSIZE See ACBRSIZE.

LACBSIZE

See ACBLSTATE.

LACOSTOPCHAR See ACBSTOPCHAR.

See ACRILING. LACSTLOG

The overall structure of the PACB is:

BASIC PACE	(FIXED)
BUFFERING Ektehsion	(VRRIABLE

The buffering extension is optional; it is present if and only if the file is accessed with buffering. There are thus two possible formats for an RCB:  $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}$ 

- 1. No buffers; the buffering extension is not present.
- PACB buffers; the buffering extension is present and the buf-fers are in the buffering extension.

If nultiple PRCB buffers exist, there uill be a buffering extension for each, immediately preceding the buffer. The basic PRCB (or NDBUF PRCB) is copied into the the RCB as words 0 through X63; an RCB "extension" is then generated in words X64 – X67. The resulting RCB thus has the following format:

G.00.00 6- 21

File System

	0 1 2 3 4 5 6 7 B 9 10 11 12 13 14 15					
D	i 2   COMPLETE RCB SIZE	-   D				
1	FILE NUMBER	1				
2	FILE NAME - 1ST CNRR.   FILE NRME - 2HD CNRR.	2				
3	FILE NAME - 3RD CHRR.   FILE NAME - 4TH CHRR.	3				
4	FILE HRME - 5TH CHRR.   FILE HAME - 6TH CNAR.	4				
5	FILE NAME - 7TH CHRR.   FILE NAME - BTN CMAR.	5				
6	FOPTIONS	6				
7	ROPTIONS	7				
8	Record size in bytes	10				
9	BLOCK SIZE IH WORDS	11				
10	UNUSEO	12				
11	CARRIAGE CONTROL CODE					
12	EOF PG   LN  ST   FK   TC   TB   8B   CAR OB   EOF T   EDF M					
13	C    TE  IC  Q     TERMINAL STOP CHARRCTER					
14	ERROR CODE					
15	LAST I/O TRANSMISSION LOG	17				
16	5   FILE POINTER					
17		21				
18	CURRENT VARIABLE BLOCK NUMBER					
19		23				
20	RECORO TPANSFER COUNT	24				
21		25				
22	BLOCK TRANSFER COUNT	26				
23		27				
24	HIGHEST BLOCK HUMBER STARTED	30				
25		31				

G.00.00 6- 22

File System

26	I FCB VECTOR					
27						
28	TOTAL HUMBER OF LACB'S					
29	IBK   DEVICE TYPE   LAST LOGICAL I/O STATUS	35				
30	LDGICAL DEVICE MUMBER	36				
31	PF   HIT    CURRENT BUFFER; TAPE DISPLACE   HO. BUFFERS					
32	CURRENT RECORD WORD INDEN	40				
33	BUFFER SIZE	41				
34	VIRTUAL LOGICAL DEVICE ND.	42				
35	FMRVT INDEN	43				
36	NUMBER OF IMPUT LACB'S					
37	NAME TYPE   FILE DISPOSITION	45				
3B	ACCESS BIT MAP   BLOCKING FACTOR	46				
39	S   h   Q   R   D     AE  RU ABR  HE  SEDFS  EDFS	47				
40	SPDOLEO DEVICE TYPE   SPODLED DEVICE AECORO SIZE	50				
41	SPOOLED DEVICE FOPTIONS	51				
42	SPOOLEO DEVICE AOPTIONS	52				
43	IDD OR ODD INDEX	53				
44	NO NOTE OTHER PROPERTY.	54				
45	NO-WAIT DISK ADDRESS	55				
46	UNUSED	56				
47	NO-MPIT LOGICAL DEVICE	57				
4B	MAD HOLD DI CONTRACTOR	60				
49	P1P2 USED BY FDEVICECONTROL	61				
50	UNUSED	62				
51	UNUSED	63				
,		-				

File System

The above words, 0-X63, are physically located in the PRCB of the file. Below, words X64-X67, are used by file system intrinsics- and are placed onto the stack by the procedure LDC\*RCB when locking the RCB. Therefore, the buffering extension, if pres- ent, will immediately follow word X63 of the actual RCB in the Control Block Table of the file.

52	DST RELATIVE OFFSET TO PACB	64
53	DST RELATIVE DEFSET TO LRCB	65
54	DST RELATIVE OFFSET TO ACB IH THE STACK	66
55	STACK RELATIVE OFFSET TO DB	67

The following identifiers are used when referring to an RCB:

(RCBSIZE)	=	ACB. (2:14)#,	size in words
ACBFNUM		ACB(1).(8:B)#,	file number
RCBNAME		RCB(2)#,	file name
ACBHRME1		ACBOBL(1)#,	file name - first half
ACBNPhE2		ACBOBL(2)#.	file name - second half
ACBFORTIONS		RCB(6)#,	FOPTIONS
ACBROPTIONS		RCB(7)#,	AOPTIONS
ACBRSIZE		ACB(B)#,	record size (bytes)
ACBBSIZE		ACB(9)#,	block size (words)
Spare		RCB(10)#.	Unused
ACBCTL		ACB(11)#,	carriage control word
ACBLSTATE		ACB(12)#,	local state flags
RCBEOF		ACBLSTATE. (1:1)#,	end of file sensed
PCBLPCTL	=	ACBLSTATE. (2:2)#,	page and line control
PCBPAGECTL	=	ACBLSTATE. (2:1)#,	page control
RCBLINECTL	=	ACBLSTRTE. (3:1)#,	line control
ACBSTREAM		RCBLSTRTE. (4:1)#,	stream I/O
RCBFKEYS	=	ACBLSTPTE. (5:1)#,	restore function keys
ACBXMITCRLF	Ξ	ACBLSTATE. (6:1)#.	transmit CR, LF to user
RCBTBLOCK	=	RCBLSTATE. (7:1)#,	disable block mode
ACBBINARYIO	=	RCBLSTPTE.(B.1)#,	B-bit terminal transfers
PCBCARRIAGE		RCBLSTATE. (9:1)#,	carriage control flag
(ACBDEFBLOCK)	=	PCBLSTATE. (10:1)#,	default blocking
PCBREADCODE	=	ACBLSTRTE. (11:4)#,	input EOF check
PCBREADTYPE	=	ACBLSTATE.(11:2)#,	input EOF type
PCBREADMOOE		ACBLSTRTE. (13:2)#;	input EOF mode
RCBMODW		ACB(13)#,	node word
PCBMODE	Ξ	ACBMOON.(0:8)#,	node setting
PCBCIROVERFLOW	j=	ACBMODN.(0:1)#,	Signifies CIR overflow
ACBSETMOOE	=	ACBMODN. (4:4)#,	FSETMODE bits
ACBTAPEERROR	÷	ACBMODN.(4.1)#.	report recovered tape error
ACB1NHIBCRLF	=	HC6MODW. (5:1)#,	inmibit terminal CR/LF
PCRQUIESCE		ACBM00N.(6:1)#,	critical output verify
ACBSTOFCHAR	Ξ	RCBMCDN. (B:8)#,	terminal stop character

File System

ACBERROR	=	ACB(14)#,	error code
ACBTLOG		ACB(15)#,	last I/O transmission log
ACBFPTR	=	ACBOBL(08)#,	current record number
acbblk		ACBOBL(09)#,	current variable block
ACBRTFRCT		ACBDBL(10)#,	logical record TFA count
ACBETFRCT		ACBDBL(11)#,	block transfer count
ACBHIBLK		ACBDBL(12)#,	highest block started
ACBFCBV		ACBOBL(13)#,	FCB Vector table entry
ACBSNONT	=	ACB(28)#, ACB(29)#,	# of LACBs
ACBSTATU		ACBSTATU. (1:1)#,	access class, status, etc. break (\$STDIN/LIST only)
ACBEREAK ACBDTYPE	-	ACBSTATH. (2:6)#,	device type
ACBACCCL	-	RCBSTRTH. (2:3)#,	device access class
ACBSUBCL		ACBSTATH. (5:3)#,	device sub-class
ACBSTATUS		ACBSTATH. (8:8)#,	last logical I/O status
ACBOSTATUS		ACBSTATU. (8:5)#,	qualifying status part
ACBGSTATUS		RCBSTATU. (13:3)#,	general status part
ACBDADDA	=	RCB(30)#,	Ldev number of file
ACBBUFX		RCB(31)#,	buffer data & Hisc. flags
ACBPAIV		ACBBUFX.(0:1)#,	privileged access only
ACBNIT		ACBBUFX.(1:1)#,	buffer hit flag
ACBCUAABUF		ACBBUFK. (4:4)#,	current buffer nor.
ACB#UMBUFS		ACBBUFK. (12:4)#,	number of buffers less 1
ACBBUFUSED		ACB(32)#,	used block word count
ACBBUFSIZE	-	ACB(33)#,	buffer size (words)
ACBSPVDEV		ACB(34)#,	spooled virtual device FMRVT index
ACBEMAVTK		ACB(35)#, ACB(36)#,	#unber of input LACB's
ACBSHENTI# ACBONTO		ACB(37)#,	type & disposition
ACBO#TYPE		ACBONTD. (0:8)#,	name type for dir. search
ACBDISP		ACBONTO. (8:8)#,	file disposition
ACBAMLD		RCB(38)#,	access mask & LDEV
ACBACCESS		ACBAMLD.(0:8)#,	access mask
RCBBLKFRCT		ACBAMLD.(8:8)#,	Blocking factor of file
ACBGSTN	=	ACB(39)#,	spool control flags
ACBSP00LED		ACBGSTW.(0:1)#,	spooled device flag
ACBSP00LI0		ACBGSTW.(0:2)#,	spooled I#/OUT
ACBSPSQ		ACBGSTW.(2:2)#,	squeeze flags
ACBSPSQZ	=	ACBGSTW.(2:1)#,	file squeezed
ACBSPRSQ		RCBGSTW.(3:1)#,	request to squeeze
RCBSPOSQ		RCBGSTH. (4:1)#,	squeeze just done
ACBNOWRITEOF		ACBGSTW.(8:1)#,	EOF advanced? last I/O: O=read, 1=urite
ACBABOATREAD		ACBGSTW.(9:1)#, ACBGSTW.(10:1)#,	abort broken re-read?
ACBNEWEOF		ACBGSTW.(11:1)#,	EOF advanced - tape file
ACBSAVEEOFS		ACBGSTH.(12:2)#,	for saving ACBEOFS
ACBEOFS		ACBGSTH. (14:2)#,	EOF flags - : EOD/:
ACBSPTYRC		ACB(40)#,	spooled dev type/recsize
ACBSPTYPE		ACBSPTYRC. (0:6)#,	spooled dev type
ACBSPREC		ACBSPTYRC. (6:10)#.	spooled dev rec size
ACBSPFOPT		ACB(41)#,	spooled dev FOPTIONS
ACBSPAOPT		ACB(42)#,	spooled dev AOPTIONS
ACBSPKDOK	=	ACB(43)#,	IDD/ODD index
ACBNOWRITDA	=	ACBOBL(22)#,	Nouait disc address
		G.00.	
		6- 2	to C

Spare = ACB(46)\*, ACBNOWRITLDEV = ACB(47)\*, ACBP1P2 = ACB(08L(24)\*, ACBP1 = ACB(48)\*, ACBP2 = RCB(49)\*; Unused #oHait logical device Used by FDEVICECONTROL

Discussion:

This flag is used to abort a broken terminal re-read. The flag is set via the RBORT parameter to FUNBREAK. If the flag is set then the READ PENDING message will be aborted along with the re-read. This feature is needed to hand-ACBABOA TAEAO along with the re-read. This featur le the BREAK...: ABORT, etc. situation.

ACBACCCL

0 - direct (e.g. disc)
1 - serial input (e.g. card reader)
2 - parallel input/output (e.g. terminal)
3 - serial input/output (e.g. magnetic tape)
4 - serial output (e.g. line printer)

ACBACCESS

(0:1) - unused (1:1) - unused (2:1) - read (3:1) - append (4:1) - urite (5:1) - lock (6:1) - execute (7:1) - save

This access security is determined by the  $\ensuremath{\mathsf{ACCCNECK}}$  intrinsic and enforced by the file system.

This is the AOPTIO#S in effect for this file access. ACBADPTIONS

This bit controls full eight bit transfers on the 2644 page mode terminal. It is adjusted by FCONTROL(26) and FCONTROL(27).

This is the block number of the current variable record format block. Rpplicable if the record format is

This is the blocking factor for the file. It is the number of records in a block. Legal values range from 1 to 255. ACBBLKFACT

G.00.00 6- 26

File System

This is the break mode flag. It is applicable if the RCB is for \$STDIN or \$STDLIST. If set it neans that the BRERK key has been hit and that the CI should have high priority access to the RCB. The flag will be cleared when a RESUME or ABORT is issued.

ACBBSIZE This is the block size, in words, of the file,

BCRBAFAK

ACBCARRIAGE

ACBOEFB LDCK

ACBOISP

ACBBTFACT This is the total number of blocks transferred to and from the file. The initial value is 00.

This is the word index, relative to the base of the block, for the selected record within the block. This is applicable if the file access is buffered. ACRBUFUSED

> This bit signifies that the file has carriage control. It ints bit signifies that the file has carriage control. It is the same as the carriage control bit in ACBFOPTIONS if the file is spooled. If not spooled, the bit is zero, and IDMOVE will pass the FURTIE carriage control parameter directly to the driver rather than embedding at as the first character of the output record.

This is the CONTROL parameter from the last FWRITE. This value is pertinent if the file was opened with carriage control. ACBCTL

This is the buffer number (O-relative) containing the most recently referenced record. Applicable if the file access is buffered. ACBCURABUF

This is the logical device number of the file. For a disc file this is the logical device number of the first extent. ACBDRODA

This bit signifies that the file is to be accessed with default blocking. The bit is initialized from the FOPEN stateword STATE. It does not need to be in the RCDB; it is nentioned here only to signify that the bit is effectively used due to the way ACBLSTATE is initialized from STOIF tively use from STATE.

This is the file close disposition derived from the FOPE% call. The only way this can be specified is via a file equation. The legal values are the same as those for FCLOSE. RCBONTYPE This is the file reference format type number and is derived from the FOPEN call. The following are legal values:

0 - full name

1 ~ account name absent 2 - group and account name absent 3 - null name

G.00.00 6- 27

This information is needed by FRENAME.

File System ACBDTYPE

ACBBINARYIO

ACBBLK

0 - moving head disc 1 - fixed head disc 7 - foreign disc 10 - card reader 11 - paper tape reader 20 - terminal

20 - terminal 24 - card reader/interpreter/punch 26 - SSLC 27 - programmable controller

26 - SSLC
27 - programmable controller
30 - nagnetic tape
31 - serial disc
40 - line printer
41 - card punch
42 - paper tape punch
43 - CRLCDMP 500 plotter
45 - CRLCDMP 500 plotter
45 - CRLCDMP 700 plotter

ACBEOF This bit is set when EOF has been sensed.

ACBEOES This is the type of EOF detected on \$STDI#(X). This field consists of two bits:

(0:1) - super colon (i.e. EOF for \$STOINX) (1:1) - regular colon (i.e. EOF for \$STOI#)

Applicable for multi-access to \$STDI#(K) only.

This is the error number for the file. It is used by all intrinsics except FOPEN. When an error is detected the error number is placed in this cell. The error number is cleared at the beginning of each callable intrinsic except FCHECK (which reads it). ACBERROR

This is the FCB vector for the file. Applicable only to disc files.

This bit controls the definition of the f1 and f2 function keys on the 2644 page node terminal; it is adjusted by FCONTROL(32) and FCONTROL(33). (Obsolete function)

File number, range from 1 to 255. Used mostly for calling routines that access things such as labels by file number. ACBENUM

ACBFORTIONS This is the FOPTIONS in effect for this file access.

This is the sequential access record pointer; it contains the next sequential record number. The initial value is OD. This value is used only by the FRERD, FURTIE intrinsics. Nouever the value is ACBEPTR

**ACBEKEYS** 

File System File System maintained by all data transferring file system ACBMODE These are miscellaneous mode flags. The constituent bits are described individually. intrinsics. **ACREMOVTY** This is the entry index into the file multi-access vector table (FMRVI). This is valid if the file access is multi-access. This is the local file name. The name is eight bytes in length with trailing blanks added. **REBHRME** This flag when set indicates that a new tape mark should be unitten before the tape is rewound or backspaced. Applicable only to magnetic tape files. ACRNEUEDE These are miscellaneous state flags. These are "global" in nature in that they are the same for all accessors in a nulti-access environment. The constituent bits are described individually. **RCBGSTRTE** ACONOURITEDE This bit is used to save the value of the local EOF advanced flag NEWEDF in IOHOWE between the I/O initiation and I/O conpletion calls. This flag is applicable if the file is accessed in nowait I/O node. ACBGSTATUS This is the general part of the last I/O status for the file. The following are the legal values: 0 - pending 1 - successful 2 - end of file This cell is used to save the I/O mode between nowait I/O initiation and completion calls. If the bit is set then the last I/O request was a write; otherwise it was a read. This cell is pertinent if the file is accessed in nowait I/O mode. **ACBNOWAITHOOE** 3 - unusual condition 4 - irrecoverable error This is the highest block number for which an anticipatory read has been issued, and is applicable if the file access is buffered. The initial value is -10. ACBHIBLK ACBNUMBUES This is the number of buffers, less one, used for the file access. Applicable if the file access is buffered. This is the page control bit. If not set then a page is assumed to consist of 60 lines (auto page eject); if set then a page is assumed to consist of 66 lines (no auto page eject). His is used primarily for line printers but is also valid for terminals; these are the only devices for which this is valid. This bit is adjusted by FCOMTROL(1) and FWATIE with the appropriate carriage ACBPAGECTL This is the buffer hit flag. If set it indicates that the last read or write request was serviced without any physical I/O required. This flag is used only for performance measurement. The code which manipulates it is optional to the file system, and is controlled by compiler toggle X3. RCBHIT This bit controls the termination of lines written to the terminal. If not set then each line is terminated with a CR and LF; if set then no line termination characters are used. This bit is valid if the file is a terminal file; it is adjusted by FSETMODE. ACBINHIBOALE ACBPATY This flag when set indicates that the file is privileged in that it has a negative file code; the user must be in privileged mode to access it. This is the line control bit. If not set then each line is post-spaced; if set then each line is prespaced. This bit is used by line printers and terminals only. It is adjusted by FCONTROL(1) and FWRITE with the applications. **ACBLINECTL** ACBOSTATUS This is the qualifying part of the last I/O status for the file. The values are unique for each general status part. See I/O Systen IMS for all legal values. This bit controls critical output verification. If set, buffered output is guaranteed to have been written to the device when control is returned to the user. This bit is adjusted by FSETHODE. propriate carriage control. **ACBQUIESCE ACBLPCTL** This are the line and page control bits, which are described separately. These are miscellaneous state flags. They are "local" in nature in that they may be different for each accessor in a multi-access environment. Bits (9:6) are initialized from the stateword local variable called STATE in FDPEN; the ten remaining bits are initialized individually. The constituent bits are described individually. ACBLSTATE This field consists of the input EOF checking type and mode, and is used to generate the P1 parameter to RTTACHIO. These fields are described individually. ACBRERDCODE

G.00.00 6- 29

File System

ACBREADHODE

This field controls the input EOF checking type. It is 01 for JOBs, 10 for SESSIONs, and 00 for OATA. REBSPVOEV This is the logical device number of the spooled device. Applicable if the file access is to a spooled device. This is the file's record size in positive bytes. This is the index into the IOO or OOO for a spoolfile. Applicable if the file access is to either a spooled device or a spoolfile. ACBSPXOOX **ACBSTATUS** 

This is the last I/O status for the file. It comes from the I/O status part of the IOCB returned by ATTACHIO. Not all ATTACHIO calls update this cell.

G.00.00 6- 30

This field controls the input EOF checking mode. It is 00 for reading \$STOIN, 01 for reading \$STOINX, and 10 for the command interpreter.

ACBSTOPCHAA This is the record termination character used for terminal reads. This character can be changed via minal reads. FCONTROL(25).

This bit signifies inter-block garbage for disc files. If set, the block size is a multiple of 128 words and therefore there is no garbage data between blocks. This fact is used to improve nultirecord I/O by mapping the request into as few RITACHIOs as possible. ACBSTAEAN

This is the sub-class part of the device type number. The sub-class is unique for each access class. The following are the legal sub-class values for each device class: ACBSUBCL

> 0 - direct 0 - direct
> 0 - moving head disc
> 1 - fixed head disc
> 7 - foreign disc
> 1 - serial input
> 0 - card reader
> 1 - paper tape reader
> 2 - parallel input/output
> 0 - terminal
> 4 - card reader/punch
> 6 - SSLC
> 7 - programmable controller
> 3 - serial input/output
> 0 - nagmetic tape 0 - nagnetic tape 7 - serial disc

4 - serial output
0 - line printer
1 - card punch 2 - paper tape punch 3 - CRLCOMP 500 plotter 4 - CRLCOMP 600 plotter 5 - CRLCOMP 700 plotter

File System

**ACAREANTYPE** 

ACBASIZE

**RCBATFACT** This is the total number of records transferred to and from the file. The initial value is 00.

ACBSAVEEOFS

**ACBSHCNT** This is the total number of LACBs that exist for this PACB. Valid if the file access is multi-access.

This is the total number of input-only LACBs that exist for this PACB. Valid if the file access is ACBSHCHTIN multi-access.

**BURSHUNTS** This is the total LACB and total input-only LACB counts, each of which is described separately.

This is the size, in words, of the ACB. The complete size (including buffers) may be calculated from the OST size containing the ABC. It does not include the buffering extension, if present. ACBSIZE

This is the AOPTIONS for the spooled dampplicable if the file access is to a spooled device. ACBSPROPT device.

This is the FOPTIONS for the spooled device. Rpplicable if the file access is to a spooled device. **ACASPEART** 

ACBSPOOLEO This is the spooled device flag. If set then the  $% \left( 1\right) =\left( 1\right) +\left( 1\right) =\left( 1\right) +\left( 1\right) =\left( 1\right) +\left( 1\right) =\left( 1\right) +\left( 1\right) =\left( 1\right) +\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left( 1\right) =\left$ 

This field is a combination of the spooled device flag and the input/output node of the spooled device. Legal values **ACBSPOOLIO** are:

00 - not spooled 01 - illegal 10 - input spooling 11 - output spooling

ACBSPAEC This is the record size, in bytes, of the spooled device. Applicable if the file access is to a spooled device.

ACBSPTYPE This is the device type (from the LDT) of the spooled device. Applicable if the file access is to a spooled

**ACBSPTYAC** This cell contains the spooled device type and record size, which are described separately.

#### File System

This bit controls the reporting of recovered magnetic errors. If not set the recovered errors are not reported to the user; if set then recovered errors are reported to the user by returning CCL and error number 39. Valid if the file is a magnetic tape file. This bit is **ACBTRPEERROR** 

adjusted by FSETHODE.

This bit controls block mode transfers on the 2644 page mode terminal. This bit is adjusted by FCONTROL(28) and FCONTROL(29). ACBTBLOCK

This is the last I/O transmission log for the file. It cones from the I/O transmission log part of the IOCB returned by RITACNIO. Not all RITACNIO calls update this cell. ACBTLOG

This is the volume table index Applicable if the file is a disc file. RCBYDRODR for the file.

This bit controls CR and LF insertion into the user buffer on the 2644 page mode terminal. This bit is adjusted by FCONTROL(30) and FCONTROL(31). ROBKMITORLE

File System

If present, the PRCB buffering extension contains from one to sixteen block buffers each having the following format:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 O BLKIOOK BLK LOEV NUMBER I I UIRIDIHIMIP 1 BLKFLRGH IOCB - STRTUS 2 BLKLSTRT IOCB - TRANSMISSION LOG 3 BLKTLOG 4 BLKBLOCK BLOCK NUMBER 6 BLKDAODR BLOCK SECTOR ADDRESS 8 BLKEXTBRSE BLOCK EKTENT BASE 9 BLOCK EKTENT SIZE 10 BLKEKTSIZE UNUSED 12 BLKBUFFER BUFFER

Other identifiers used:

BLKIOPEND

BLKFLAGU BLKFLAGS BLKUNALLOCEKT BLKREVERSE BLKODNTWAIT BLKIOOUT BLKOIRTY

= BLK(1)#, Flag and LDEV word
= BLKFLRGU.(0:8)#, block logical device number
= BLKFLRGU.(0:8)#, block I/O flags
= BLKFLRGU.(10:1), Block from unalloc. extent
= BLKFLRGU.(10:1), Block from unalloc. extent
= BLKFLRGU.(11:1), FRENDBRCKWARD (not used )
= BLKFLRGU.(13:1)#, last I/O was urite?
= BLKFLRGU.(14:1)#, buffer nodified?
= BLKFLRGU.(14:1)#, buffer nodified?
= BLKFLRGU.(14:2)#, I/O complete - not dirty
= BLKFLRGU.(14:2)#, I/O complete - not dirty
= BLKFLRGU.(14:2)#, I/O complete - not dirty

BIKTOCOMP BLKIOCB

G.00.00 6- 34

G.00.00 6- 33

File System

Discussion: BLKBLDCK

This is the block number of the data contained in the buffer. R value of -10 indicates that the buffer is енрty.

This is the actual file system buffer space. Each buffer is exactly one file block in size. BLKBUFFER

BLKDRDDR This is the block's logical device and sector number.

This flag is set if the contents of the buffer has been modified. When the block buffer is re-used this flag is checked to see if the block needs to be written to the BUKDIRTY

device.

This bit will be on if the I/O was already completed via "DONT'WAIT" but the status has not been checked yet. Check the status before using the block in the buffer. BLKOONTWAIT

This is the sector address of the extent base in which the block resides. This is used for disc caching. BLKEXTBRSE

The size, in sectors, of the extent in which the block resides. This is used for disc caching. BLKEKTSIZE

These are the miscellaneous flags associated with the block, which are described separately. BLKFLAGS

BLKTOCK

This is the IOCB returned by the I/D system when the block I/O has completed. On a blocked I/O request this is obtained from the RTIRCHID call; on an unblocked I/O request this is obtained from WRITFORIO.

This is the buffer modified flag (BLKOIRTY) and the I/O in progress flag (BLKIOPEND), which are described separately. This field is usually interrogated to see if it contains the value 2, which means that the buffer has been modified but not yet written to the device. BLKICCOMP

This is the mode of the I/O operation for the block. It is set by a write and cleared by a read. BLKIOOUT

This is the I/O in progress flag. It is set if the I/O is pending; it is cleared when the I/O has completed. BLKIGPEND

This is the IOO index of the unblocked I/D request for the block. It is used as the argument to WRITFORIO, which ensures the completion of the I/D request. RUKTOOX

This is the logical device number of the block. (Valid only for disc files.)  $\,$ BLKLDEV

File System

BLKTLOG

BLKUNALLOCEXT

The I/O status part of the IOCB consists of the PCB number and the error code for the completed I/O  $\,$ BLKLSTAT ber and the request.

The transmission log part of the IOCB is the number of words or bytes transferred by the the I/O request.

BLKREVERSE

This bit would indicate that we are reading back- wards from a tape. Nowever, currently FREROBRCK- WARDS can only be performed unbuffered.

This bit signifies that the block was "read" from an unallocated extent. Actually, the buffer was simply cleared with fill characters. Therefore, if a write is attempted to the block residing in this buffer, it must pass through FCONVBLK to allocate the extent first.

## File Control Block (FCB)

The FCB coordinates access to a file on a sharable device. At present the only sharable device is a disc, so only disc files have FCBs.

The information contained in an FCB is derived from the file label. The FCB is used to hold this information, rather than the file label, since it can be accessed more quickly.

There are two strategies to choose from in deciding where to place the FCB. If the file has been opened exclusive and no other process could possible share this file, then the FCB is placed into the PKFILE area (or in a MOBUF expandable CDI if it won't fit in the PKFILE area or if the program is run with MOCB). If the file could possible be shared, then the FCB is always placed in a shared control block table. The number of a data segment containing a list of shared file system data segments is kept in system global location 1076 cotal. The size of the FCB depends on the maximum number of extents specified at FOPCH; there are 44 (octal) words plus two re extent. There will be at least one extent, since the file label always exists in the first extent. The FCB extent map is in terms of logical device and sector number. The extent map in the file label is in terms of volume rather than logical device; the map is converted by VTABDLDEV when the label is read, and converted back by LDEVIDVIRB when the label is written to disc.

The FC8 has the following format:

	0 1	2	3	7	Ð		12	13	14	15		
0	1	1		COM	PLETE	FCB	SIZ	Ē			0	
1	İ			SPA	AE	•					1	
2				FOP	TIONS						2	FCBFOP-
3				OEVICE SP	ECIFIC	ATIO	N				3	TIDMS FCBOEVICE
4	PREV.	LBC	K  (	OEV. TYPE	C		101	VIC	SU	BTYPE	4	
5				NO. OPEN	SFOR	DUTP	UT				5	
6	i		<u>-</u> -	NO. OPEN	S FOA (	RNY	1001				6	
7				AIN A	IUMBEA			<b>-</b>			7	FCBRIN
θ				EKC LUS:	IVE STA	TUS			· <b>-</b>		10	
9	CI		ı	MVTABK		VIII:	SK		~		11	STAT FCBPVINFO
10				FILE							12	FCBFLIM
11				rite i	T111						13	

G.00.00 6- 37

File System

12	UNUSED	14	
13		15	
14	i		FCBEOF
15		17	
16	NO. USEA LABELS WRITTEN   NO. USER LABELS AVAIL.	50	FCBUSEALBL
17	EXTENT SIZE IN SECTORS	21	FCBEXTSIZE
18	BLOCKING FACTOA   SECTOAS PER BLOCK	22	
19	SECTOR OFFSET TO ORTA   DISP   HO. EKTENTS - 1	23	
20	LAST EKTENT SIZE IN SECTORS	24	
21		25	EXTSIZE
22	GROUP NAME - 1ST CHAR.   GROUP NAME - 2ND CNAR.	26	FCBGN
23	GROUP NAME - 3AO CNAA.   GROUP NAME - 4TH CNAA.	27	
24	GROUP NAME - 5TN CNAA.   GROUP NAME - 6TN CNAA.	30	
25	GROUP NAME - 7TM CHAR.   GROUP NAME - 8TM CHAA.	31	
26	ACCT NAME - 1ST CHAA.   ACCT NAME - 2ND CHAR.	32	FCBAN
27	ACCT NAME - 3AD CHAR.   ACCT NAME - 4TH CHAR.	33	
28	ACCT NAME - 5TH CHAR.   ACCT NAME - 6TH CHAR.	34	
29	ACCT NAME - 7TN CNAR.   ACCT HAME - BTH CHAR.	35	
30		36	FCBSTART
31	STAAT OF FILE BLDCK NUMBER	37	
32		40	FCBEND
33	CURRENT NUMBER OF ORTH BLOCKS IN THE FILE	41	
34		42	FCBNUM-
35	NUMBER OF OPEN AND CLOSE RECORDS (MESSAGE FILE)	43	BPENCLSAEC .
36	LDGICAL DEVICE NUMBER	44	FCBEXTHRP
37	FIAST EXTENT SECTOR WUMBER	45	
į	•		

6.00.00 6- 38

File System

	•	!
LOGICA	AL DEVICE NUMBER	
	LAST EXTENT SECTOR NUMBER	l

## Other identifiers used:

FCBSIZE	= FCB.(2:14)#,	size in words
FC8 LKST	= FCB(4).(0:2)#,	previous lock state
FC80TYPE	= FCB(4).(2:6)#,	device type
FCBCRUNCN	= FCB(4).(B:1)#,	pending crunch disposition
FCBSUBTYPE	= FCB(4).(12:4)#,	device subtype
FCBOCNTOUT	= FCB(5).(0:B)#,	no. accessors - output
FCBOCNT	= FCB(5).(B:B)#,	no. accessors
FCBCLASSFLG	= FCB(9).(0:1)#,	PV class flag
FCBMVTABX	= FCB(9).(4:4)#,	Mounted volume table index
FCBVMASX	= FCB(9).(8:8)#,	Volume Mask
FCBLBLEDF	= FCB(16).(0:8)#,	no. labels written
FCBLBL	= FCB(16).(B:B)#,	no. labels available
FCBBLKFRCT	= FCB(1B).(0:B)#,	blocking factor
FC8SECTPBLK	= FCB(1B).(B:B)#,	sectors per block
FC8SECTOFF	= FCB(19).(0:B)#,	sector offset to data
FCB0ISP	= FCB(19).(B:3)#,	pending disposition
FCBNUMEKTS	= FCB(19).(11:5)#,	no. extents less 1
FCBOCHTIN	= FCB(21).(B:B)#,	no. accessors - input
FCBLASEL	= FCB08L(18)#,	label LOEV and sector
FCBLDEV	= FCB(36).(0:B)#,	label LDEV

## Oiscussion:

FCBACBOST	This tine	is	the the	OST of FCB.	the ACB This	tha	ssu f	cre	ated	at	the	Same
					locating	the	FCB.	Tu	conj	uncti	.un	HITH

CBACEV	This is created junction	at	the sam	e time	as the	FCB.	This	is	used in	иаs con-
--------	--------------------------------	----	---------	--------	--------	------	------	----	---------	-------------

FCBAN	This	18	the	account	пане	of	the	file.	Ιt	is	eight
	bytes	in	length	with trai	ling b	lank	s ado	led.			

	·
FCBDEVICE	This specifies the device on which the file resides. If
	it is positive them it represents a logical device rumber:
	of negative it represents a (negative) device class index.

## File System FC80ISP

FCBGN

This	is	the	pending	FCLOSE	disposition	for	the	file.
Legal	valu	es are						

٥	_	no	change	

FCBCRUNCN This bit governs if space will be returned beyond the EOF upon the last FCLOSE of the file.

# 0 - no change 1 - return space beyond EOF

FC80TYPE This is the device type number of the first extent of the file. See ACBOTYPE for a list of legal values.

FCBEND Block number of the file's EOF, relative to FCBSTAAT. FCBEOF

This is the end-of-file pointer for the file. It is a double integer representing the number of records in the file. It can also be viewed as the record number of the next record past EDF.

This is the exclusive status of the file access. If -1 then the file is being accessed exclusively; otherwise it is the number of semi-exclusive accessors. **ECRENCISTAL** 

FCBEXTMAP This is the extent map of the file. The number of extents is specified by FCBMUMEKTS; a OD extent descriptor indicates that the extent has not been allocated.

This is the extent size, in sectors, of the file. All extents in the file except possibly the last have this size. This is a logical value, and legal values range from 1 to 65535 sectors. This restricts the maximum file size to 2097120 sectors (268,431,360 words). FCBEXTSIZE

FCBFLIM This is the end-of-space pointer for the file. It is a double uprd integer representing the maximum number of records (fixed length record format) or blocks (undefined or variable length record format) in the file.

FORFORTIONS This is the FOPTIONS in effect for the file.

This is the group name of the fale. It is eight bytes long with trailing blanks anded

FOBLOSSI This is the logical device and sector number of the file late1, which is the same is the first extent descriptor.

<sup>0 -</sup> no change 1 - save permanent 2 - save temporary and rewind 3 - save temporary but do not rewind

<sup>4 -</sup> release 7 - invalid file (file label access error)

File System File System FUBLRSTEXISIZE

This is the size, in sectors, of the last extent in the file. If the file has one extent then this is the same as FUBEXISIZE; otherwise this value may be different from FUBEXISIZE. This is the size of the last physical extent for the file; it is not the size of the last allocated This is the number of accessors for the file. Riternatively it can be viewed as the number of PACBs created for the file. FCBOCHT FCBOCHTIN This is the number of file accessors having input access. This is the number of user labels allocated for the file. Since each label is a sector long, this is also the number of sectors allocated for user labels. FCBLBL FCBOCHTOUT This is the number of file accessors having output This is the RIN number used to support dynamic locking (i.e. FLOCK and FUNLOCK) for the file. If there is no FCBRIN This is the end-of-data pointer for the user labels. It is analogous to FCBEOF in that it represents the number of labels written. The initial value is 0. FCBLBLEOF dynamic locking them this number is zero. This is the sector offset from the file label to the first block of the file. This is not necessarily equal to FCBIBL+1 since an integral number of blocks are allocated for the file and user labels. FCBSECTOFF is the logical device number of the first extent of FCBLDEV FCBLKST This is the previous lock state of the file and is derived from the file label. Legal values are: FCBSECTPBLK This is the number of sectors in a block for the file. This is the size, in words, of the complete FCB. It includes the extent map. 0 - no accessors FCBSIZE Block number of the file's start, excluding the file label block.  $% \label{eq:block}%$ **ECRSTART** If the file resides on a private volume, then this field represents the mounted volume table index of the volume set entry on which the file resides. FCBMVTRBX FCBSUBTYPE This is the device subtype number of the first extent. This field describes the user labels for the file. It consists of FCBLBL and FCBLBLEOF, described separately. FCBUSERLBL This is the OST of the new FCB for the file. It is used in conjunction with FCBRCBOST to move the FCB to a system (shared FCB) control block table when the second accessor is established. If this value is zero then there is no new FCB; if nonzero then a new FCB has been FCBNEWFCBDST If the file resides on a private volume set, this bit mask signifies which volume of the set in which the file resides. But 15 is on it resides on the first volume, bit 14 if on the second, etc. FCBVMRSK This is the vector table entry of the new FCB for the file. It is used in conjunction with FCBRCBV to move the FCB to a system (shared FCB) control block table when the second accessor is established. If this value is zero then there is no new FCB; if nonzero then a new FCB has been centred. FCBNEWFCBV been created. This is the maximum number of extents, less one, alloued for the file. It is not the number of extents presently allocated, which is always determined by counting nonzero entries in the extent map. FCBNUMEXTS FCBNUMOPENCLSREC Number of open and close records in the message file.

\_\_\_\_\_

File System

# File Label (FLAB)

## . . . . .

G.00.00 6- 41

The file label has the following format:

0 1 2 3 7	8	12	13	14	15		
FILE NAME - 1ST CHRR.	F'LE	NAME -	2ND	CHAF	.	0	FLLOCHAME
FILE NAME - 3RD CHAR.	FILE	NAME -	4TH	CHAR	.	1	
FILE NAME - 5TH CHAR.	FILE	NAME -	6TN	CNAF	.	2	
FILE NAME - 7TH CHAR.	FILE	NRME -	8TH	CHA	.	3	
GROUP NAME - 1ST CHAR.	) GROUP	NAME -	2N0	CHAF		4	FLGRPNAME
GROUP NAME - 3RO CHAR.	) GROUP	NAME -	4TN	CHAF	.	5	
GROUP NAME - 5TH CHAR.	GROUP	NAME -	6TN	CNAF	.	6	
GROUP NAME - 7TH CHAR.	GROUP	NAME -	8TH	CNAF	.	7	
ACCT NAME - 1ST CHAR.	ACCT	NAME -	2110	CHA	ì.	10	FLACCTNAME
ACCT NAME - 3RD CHAR.	RCCT	NRME -	4TH	CHR	ī.	11	
ACCT NAME - 5TN CHAR.	RCCT	NAME -	<b>6</b> TH	CHA		12	
ACCT NAME - 7TH CHAR.	ACCT	NAME -	8TH	CHA		13	
CREATOR NAME - 1ST CHAR.	.   CREAT	OR NAME	- 2	ND CI	AR.	14	FLUSERID
CREATOR NAME - 3RD CHAR	.   CREAT	OR NAME	- 4	TH C	IAR.	15	
CREATOR NAME - 5TH CHAR.	.   CREAT	OR NAME	- 6	TH CI	IAR.	16	
CREATOR WAME - 7TH CHAR	.  CREA1	OR HAME	- 8	TN CI	IAR.	17	
LOCKWORD - 1ST CHAR.	LOC	HORO -	2ND	CHAR	.	20	FLLOCKWORD
LOCKWORD - 3RO CHAR.	LOC>	WORD -	4TH	CHAR	i	21	
LOCKWORD - 5TH CHRR.	LOC>	WORD -	6TN	CHAR		22	
LOCKWORD - 7TH CHAR.	LOC	WORO -	8TN	CHAR		23	
\$500 PT	TY MATRI	· ·				24	FLSECMX
SECORI		.n	<b>-</b> -			25	
FILE LANGUAGE ATTRIBUTE	I		Ī	SR	S	26	
I					1		

File System

## File Label (Cont.)

G.00.00

<u>File Label (Cont.)</u>	
CREATION DATE	27 FLCREATE
LAST ACCESS DATE	30 FLLASTACC
LAST MODIFICATION DATE	31 FLLASTMOD
FILE CODE	32 FLFILECODE
C     MVTABX   VMASX	33 FLPVINFO
S   R   L   X   SUBTYPE   DISC TYPE   R/W	34 FLLOCK
NO. USER LABELS WRITTEN   NO. USER LABELS AVAIL.	35 FLUSERLBL
	36 FLFLIM
FILE LIMIT IN BLOCKS	37
	40 FLFCBVECT
FCB VECTOR	41
	42 FLCHECKSUM
COLD LOAD ID	43 FLCLID
FOPTIONS	44 FLFOPTIONS
RECORD SIZE IN BYTES	45 FLRECSIZE
BLOCK SIZE IN WORDS	46 FLBUKSIZE
SECTOR OFFSET   ND. EXTENTS -1	47
LAST EXTENT SIZE IN SECTORS	50 FLLASTEXT-
EXTENT SIZE IN SECTORS	SIZE 51 FLEXTSIZE
	52 FLEOF
END OF DATA POINTER	53
VOLUME TABLE INDEX	54 FLEXTMAP
1ST EXTENT SECTOR NUMBER	55
	I

G.00.00 6- 44

G.00.00 6- 43

	File System	File System						
File Label (Cont.)   VOLUME TABLE INDEX    LAST EXTENT SECTOR NUMBER	-	FLRESTORE (FLICARD) FLEXCL FLSR FLSR (FLSRLX) FLSUBTYPE FLSTATUS (FLUBLOF) (FLUBL) FLSECTOFF FLWHEXTS	= FLRB(2B).(1:1)N, file being restored = FLRB(2B).(2:1)M, File loaded = FLRB(2B).(0:2)M, File loaded = FLRB(2B).(0:2)M, S. R. bits = FLRB(2B).(0:4)M, S. R. i. bits = FLRB(2B).(0:4)M, S. R. i. bits = FLRB(2B).(4:4)M, device type = FLRB(2B).(4:4)M, device type = FLRB(2B).(14:2)M, urite/read status = FLRB(2B).(14:2)M, urite/read status = FLRB(2B).(18:B)M, no. labels available = FLRB(3B).(18:B)M, no. extents less 1					
FILE ALLOCATION TIME	-   154 FLALLOCTIME	FLLABEL Flytab	= FLRBOBL(22)#, label YTAB and sector = FLRB(44).(0:8)#, label YTAB index					
	155	Discussion:						
FILE ALLOCATION DATE	156 FLALLOCDATE	FLACCTHAME	This is the account mane of the file. It is eight bytes in length with trailing blanks added.					
	-	FLALLOCORTE	Date that the file was allocated on this system.					
START OF FILE BLOCK HUMBER	160 FLSTART	FLALLOCTIME	Doubleword containing the time that the file was allocated on this system.					
PLOCK HIMDEO OF THE OF THE	162 FLEND	FLBLKSIZE	This is the block size, in sectore, of the file.					
BLOCK NUMBER OF END OF FILE	163	FLCHECKSUM	This is the exclusive-DR checksum of the file label (ex-					
NUMBER OF OPEN AND CLOSE RECORDS (MESSAGE FILE)	164 FLNUMOPEHCLSREC		cluding words 34, 42, and 43 octal) and is used for error detection. Each time the file label is read from disc the check sum is calculated and compared against the value recorded in the file label. Similarly, each time the file					
DEVICE NAME - 1ST CNAR.   DEVICE HAME - 2ND CHAR.	174 FLOEVHAME		label is written to the disc the check sum is calculated and inserted into the file label.					
DEVICE NAME - 3RD CHAR.   DEVICE NAME - 4TH CHAR.	175	FLCLID	This is the cold load number in effect the last time that					
DEVICE HAME - 5TH CHAR.   DEVICE HAME - 6TH CHAR.	176		the file was accessed. This should always be the current cold load number. If it is not, it means that the system					
DEVICE HAME - 7TH CHAR.   DEVICE HAME - 8TH CHAR.	177		crashed while the file was open and that the data in the File label should be "reset" (principally the FC8 vector FUFCBYECT).					
Other identifiers used:		FLCREATE	This is the creation date of the file. It is in the format defined by the intrinsic CRLENDAR.					
FLSECURE = FLAB(22).(15:1)#, file secure i (FLSRRELEASE)= FLAB(22).(14:1)#, STORE/RESTORI FLCLASSFLG = FLPYINFD.(0:1)N, Class flag b:	released bit	FLOEVHAME	This is the FDPEH device specification that was used when the file was created. This information is needed when new extents are allocated.					
	ne table index	FLOTYPE	This is the device type number of the first extent of the file; see ACBDTYPE for a list of legal values. This value is determined by configuration.					

G.∞..∞ 6- 46

File System

FLELIN

```
File System
FLEND
                                                           Humber of current data blocks (that is, the end of file block number relative to the start of file).
                                                          This is the end-of-file pointer for the file. It is a double word integer representing the number of records in the file. It can also be viewed as the record number of the next record past EOF.
FLEOF
FLEXCL
                                                          This is the exclusive access flag for the File. If set it means that the file has been opened exclusively by a single accessor. If not set then the file is potentially accessible by others.
                                                          This is the extent map of the file. The number of extents is specified by FLNUMEXTS; a 0D extent descriptor indicates that the extent has not been allocated.
FLEXTMAP
                                                          This is the extent size, in sectore, of the file. All extents in the file, except the last, have this extent size. This is a logical value, and legal values range from 1 to 65535 sectors. This limits the maximum file size to 2097/20
FLEXTSIZE
                                                         If monzero, this is the vector of the FCB for the file. If zero, the File is not being accessed.  
FLECBVECT
FIFTIFCONE
                                                         This is the file code of the file. Known values are:
                                                          1024
1025
1026
1027
                                                                                             User Subprogram Library
Basic Data
Basic Program
Basic Fast Program
Basic Fast Program
Relocatable library
Program File
Segmented Library
View Form File
View Fast Forms File
View Reformat File
Cross Loader RELIT File (SAVE)
Cross Loader RELIT File (DISPLAY)
Edit Duck File
Edit KEEPD File (COBOL)
TOP DIATY File
TOP Proof Marked DIRAKED
TOP Proof Marked non-EOBOL File
TOP Proof Marked COBOL File
TOP Proof Marked COBOL File
TOP Proof Marked COBOL File
TOP Proof Marked COBOL File
TOP Proof Marked COBOL File
TOP Proof Marked COBOL File
TOP Proof Marked COBOL File
TOP Proof Marked COBOL File
TOP Proof Marked COBOL File
TOP Horifile
                                                                                                  User Subprogram Library
                                                         1028
1029
1031
1035
1036
1037
1040
1041
1042
1050
1051
1055
1056
1057
1058
1059
1060
1070
                                                                                                TDP Workfile
TDP Workfile (FORME)
RJE Punch File
QUERY Procedure File
KSOM Key File
```

G.00.00 6- 45

```
GRAPH Specification File
User Logging Log File
Self-describing File
HPURD Document
HPURD Document
HPURD Document
HPURD Document
HPURD Document
HPURD Document
HPURD Document
HPURD Document
HPURD Document
HPURD Document
HPURD Document
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication File
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HPURD Logication
HP
```

This is the end-of-space pointer for the file. It is a double integer representing the maximum number of

File System records (fixed length record format) or blocks (undefined or variable length record format) in the file. **FLFOPTIONS** This is the FOPTIONS of the file. FLGRPNAME This is the group name of the file. It is eight bytes long with trailing blanks added. This is the volume table index and sector number of the file label, which is the same as the first extent descriptor. FLLRSTRCC This is the last access date of the file. It is in the format defined by the intrinsic CRLEHORR. FLLABEL FLLASTMOD This is the last modification date of the file. It is in the format defined by the intrinsic CALENDAR. This is the size, in sectors, of the last extent in the file. If the file has one extent, then this is the same as FLEXTSIZE; if the file has more than one extent, then this value nay be different from FLEXTSIZE. This is the size of the last physical extent for the file; it is not the size of the last allocated extent. FLLASTEXTSIZE This is the number of user labels allocated for the file. Since each label is a sector long, this is also the number of sectors allocated for user labels. FLLBL This is the end-of-data pointer for the user labels. It is analogous to FLEOF in that it represents the number of labels written. FLLBLEOF This is the LOADEO flag for the file. If set, it means that the file is a loaded program or SL file and cannot be modified except by a privileged accessor. This flag is set and cleared by the loader, not the file system. FLLORO This identifies the word containing the lock bits, which are described separately. FELOCK This is the lock word of the file. It is eight bytes long with trailing blanks added. If it is all blanks, then the file does not have a lockword. FLLOCKHORD FLLOCHRME This is the local name of the file. It is eight bytes long with trailing blanks added. FLNUMENTS This is the number of extents, less one, allowed for the file. It is not the number of extents allocated. Legal values range from 0 to 31, i. e., 1 to 32 extents.

File System

FLRESTORE

FLSA

**FLPVINFO** File label private volume information. This is in the same format as the FCBPVINFO.

FLRECSTZE This is the record size of the file in negative bytes.

This is the RESTORE flag for the file. If set, it means that the file is being RESTOREd and cannot be accessed. RESTORE also sets the STORE bit for the file (FLSTORE); see FLSA for a full description of the use of these bits. This flag is set and cleared by STORE/RESTORE, not the file system.

This is the security matrix of the file. The bits are organized into five groups of six bits each. (Bits 0:2 are not used.) The groups correspond to the access types: READ, RPDEND, WRITE, LOCK, and EXECUTE. Lithin each group, each bit specifies who may have the access: RNY, RCCOUNT NGR, RCCOUNT LIB- RRRIRN, GROUP, GROUP LIBRARIAN, CREATOR. FLSECMX

This is the sector offset from the file label to the first block of the file. This is not necessarily equal to FLIB!+! since an integral number of blocks are allocated for the file and user labels. FLSECTOFF

This is the file security enforcement flag for the file. If not set, then the file has been RELERSEd and the security natrix FISECHX should be ignored. If set, then secure as specified by the security natrix. FLSECURE

This is the STORE and RESTORE flags for the file, which are described separately. STORE and RESTORE decode the two-bit field to indicate their operation. Legal values are:

0 - file not in use by either STORE or RESTORE 1 - filegal value 2 - file being STOREd 3 - file being RESTOREd

The file system interprets the leftmost bit as indicating that the file is being accessed by either STORE or RESTORE. The rightmost bit is interpreted as indicating what access should be permitted: 0 (file being STOREd) allows read access; 1 (file being RESTORED allows no access. This field is set and reset by STORE/RESTORE, not the file system.

FLSRL This is the STORE, RESTORE and LOROEO flags for the file, which are described separately.

This is the STORE, RESTORE, LOROED and exclusive flags for the file, which are described separately. FLSALX

FLSARE LEASE

This flag is used by STORE/RESTORE. If a file is STOREd with the ";RELERSE" keyword, STORE will set this flag

G.00.00 6-49

FLNUMOPEHCLSREC Number of open and close records in the message file.

File System

in the tape copy of the file label. RESTORE will allow any user to access such files, regardless of the file's normal security. If this bit is off in the tape copy of the file label, RESTORE applies normal security checks (as defined by the information in FLSECIXX and FLSECURE). This bit is zero for files on disc.

Block number of the file's start, excluding the file label block.

This is the read/write status of the file. Legal values are:

0 - no accessors 1 - read 2 - write 3 - read/write

FLSTRRT

FLSTATUS

FLSTORE

This is the STORE/RESTORE flag for the file. If set it means that the file is being either STOREd or RESTOREd. The RESTORE bit (FLRESTORE) must be interrogated to determine which operation is taking place; see FLSR for a full description of the use of these bits. This flag is set and cleared by STORE/RESTORE, not the file system.

This is the device subtype number of the first extent of the file. This value is determined by FLSU8TYPE configuration.

This is the creating user name of the file. It is eight bytes long with trailing blanks added. FLUSERID

This field describes the user labels of the file. It consists of FLLBL and FLLBLEOF, which are described separately. FLUSERLBL

This is the volume table index of the first extent of the file. FLYTRE

File Systen

## File Multi-Access Vector Table (FMRVT) DST(X54)

The FMRNT is used to locate shared PACB's for files opened multi-access. When an old disc file has been opened multi-access, the FMRNT is searched to determine if the file has previously been opened. The JITOST and the DADDR found in the FMRNT are compared to the JITOST of the job and the DADDR of the device or disc file being opened multi-access. If an entry exists for the file, then the PACB can be easily located for that file. If this is the first process opening the file, then an entry is created and inserted into the FMVAT for the file.

Spoolfiles are opened multi-access, therefore, they will have entries in the FMRVI. \$\$TOLM and \$\$TOLIST also have entries in the FMRVI since they too are opened multi-access.

## Zero Entry Format

CURRENT TABLE SIZE	O FM'CURA'SIZE
ENTRY SIZE = 6	1 FM'ENTRY'SIZE
MAXIMUM TABLE SIZE	2 FM'MAX'SIZE
0	3
0	4
0	5

## Descriptions:

FM'CURR'SIZE The current size of the FMRVT in words. This value increases in increments of X200 words until FM'MRX'SIZE is reached.

FM'MAX'SIZE The naxinum allowable size in words that the FM'CURR'SIZE can get. The current value of this is X4000. FM'MRX'SIZE can be changed only by changing the code in Initial. The open of the multi-access file is falled if this naxinum is reached.

FM'ENTRY'SIZE Size in words of an FMRVT entry, 6 words at present.

### Typical Entry Format

	0		1		2		3		6	7		8			12		13	14		15	_	
ļ	1	ī	G	I	D	Ī		1			_			U	NUSI	ED			_		٥	
										JII	(	ST									1	FM'JITOS1
į			LI	DG:	IC	11	DEV	ICI	Ε												2	FM'DADDR
									)IS	K F	DI	RES	S								3	
						•••			000		·	TDR			-+-						4	FM'PACBV
									HHL	.0 1	E	. IUK							_		5	

= FMRVT(0).(2:1)N, Device bit = FMRVT(0).(1:1)N, Global multi-access bit = FM'ORDOR(0).(0:8)N, Logical device number of file FM'DEVICE EU, FDEA

Descriptions:

The disc address of the File label for disc files. For device files, the disc address is zero.

FM'DADDA FM' DEVICE

This bit is 1 for device files and 0 for disc files.

 $\ensuremath{\mathsf{LOEV}}$  device number of device files or the LDEV of the disc containing the file label for disc files.

FM" LDEV

FM' JITOST

The DST number of the JIT for the job that has the file open. If this field is nonzero, then only processes in the family tree of this particular job can open the file. This field is zero if the file was open global multi-access.

FM'GLDBAL

This bit is 1 if the file was opened global multi-access, this allows multi-access to the file between jobs.

FM' PACEV

The PACB vector for this multi-access file. Used to easily find the Physical Riccess Control Block for files opened multi-access.

G.00.00 6- 53

G.00.00 6- 54

File System

## Shared CBT DST

In sysglobal X76 (RBSDLUTE X1076) there exists the shared Control Block Table BST number. This DST holds a list of shared CBT's. Shared CBT's are used to keep any and all file systen control blocks that have the potential to be shared between processes. Any disc file opened shared ull have its FCB kept in one of these CBT's. Also, all terminal PRCB's uill be stored in a system shared CBT so that an extra data segment is not usated. This is possible because all terminal access is performed NOBUT, which means that the PRCB uill be a minimal PRCB and can be placed in these CBTs. Lastly, any file opened uith global file access will have all its control blocks placed into these system CBT's.

The format of the system shared CBT DST is similar to a Control Block Table. It has the same words of overhead and the data (the list of DST's) starts in the next word after the overhead. The system CBT's are created one at a time as needed. Usually, there are only a few DST's in the list.

1	-1
TABLE SIZE IN WDADS (X200)	0
DST NUMBER OF THIS TABLE	1
0	-  2
0	] 3
•	4
0	5
0	6
0	7
1ST. SHRRED CBT DST NUMBER	10
2HD. SHARED CBT DST HUMBER	11
	1
:	_
118TH. SHARED CBT DST NUMBER	177
	,

### System Global Area (SYSGLD8)

The file system uses several words in the system global area for its own

SHFCBDST	æ	SYSDB+X76.	shared CBT DST no.
MONITOR	Ξ	SYSDB+X77.	nonitoring flag word
MAXSSECT	=	SYSDB+X100,	nax N spoolfile sectore
NUMSSECT	=	SYSDB+Z102,	current N spoolfile sectors
EXTSSECT	=	SYSDB+2104,	N sectors/spoolfile extent
SPOOLINDEX	=	SYSDB+X132,	class spool index
CSIONAIT	=	SYSDB+X135,	CSIDWAIT PLABEL
CCLOSEPLABL	Ξ	SYSDB+X140,	CS CCLOSE PLABEL - FPROCTERM
DSCNKPLABL	=	SYSDB+X335,	DSCHECK PLABEL
OSOPENPLABL	=	SYSD8+X336,	DSDPEN PLABEL
DSCLOSEPLABL	=	SYSDB+X337,	DSCLDSE PLABEL
SDSLDEVLABEL	=	SYSDB+X323,	PLABEL for SDSLDEV
MANUCPLABL		SYSDB+7340;	MANAGEURITECONV PLABEL
GLOBALAFTOST	=	SYSGLBEXT+2121	Global AFT DST number

## SIRs, Locks, and Deadlocks

The file system uses two SIRs: the File SIA, which is intended to protect file label integrity, and the FRMYT SIR, which is to guarantee the integrity of the FRMYT. Since the File system locks these resources and also locks control blocks, deadlocks can occur if locking is done in the urong order. Not only must the file system handle locking correctly, but the entire ensemble of the file system, its callers, and its callees must do so also These include KSBM, which has a SIR of its oun, SYSOUMP, and STDRE, which lock the File SIR because they tueak bits in file labels. The presently accepted order is: cepted order is:

Get FMRVT SIR Lock ACB Get File SIR Lock FCB

It may not be necessary to do all of these things in any particular procedure. In nodifying a procedure, you should be sure that any of these locks which you change are consistent not only within your own code, but also with its callers and callees.

#### 1

## CHAPTER 7 PROCESS TRBLES

The operating system maintains state, control, and accounting information on each process. The data structures for this purpose are the process control block table (PCB; core resident, 1 entry per process) and the process control block extension (PCB; contained in the process' stack below 01). Process related information which nust be accessible when the process' stack is not present in main memory is maintained in the process' PCB antry. RIL other process related information is maintained in the process? PCBX.

R procese is identified in the system by its PCB entry number, referred to as its PIM (process identification number), or by its PCBPT=(PIM)=(PCB entry eize).

The structure of the PCB table, PCB entry format, PCBX etructure, and PCBX format are specified in this chapter.

## Procese Control Block Table Structure and Format

## Fixed Cells Related to PCB

4 PCB relative index of current process' PCB sntry X1003 Rbsolute address of the PCB table base The bank & address are represented as per the MPEV ERS. X1271 PCB relative address of nead of dispatching queue's PCB entry X1272 PCB relative address of tail of dispatching queue's PCB entry PCB Entry O Format

Process

0	* OF CONFIGUREO ENTRIES
1	ENTRY LENGTH (X25)
S	N OF UNASSIGNED ENTRIES
3	TRBLE RELATIVE INCEX TO FIRST UMASSIGNED ENTRY
4	TABLE RELATIVE INDEX OF LAST FREE ENTRY
5	HIGH URTER MARK
6	NUMBER OF PRIMARY CONFIGURED ENTRIES (0)
7	HERO OF IMPEDED QUEUE PCB RELATIVE INDEX
8	TAIL OF IMPEDED QUEUE PCB RELATIVE INDEX
9	NUMBER OF CURRENTLY IMPEDED PROCESSES
10	NUMBER OF MRXIMUM IMPEDED PROCESSES (CURRENT)
11	CUMULATIVE NUMBER OF IMPEDED PROCESSES(CURRENT)
12	0
13	0
14	o
15	0
16	0
17	0
18	0
19	0
20	0

G.00.00 7- 1 G.∞.00 7- 2

## Process

## Unassigned PCB Entry Format

0	0
1	TABLE RELATIVE INDEX TO NEXT UNASSIGNED ENTRY
2	0
3	0
4	0
5	0
6	•
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	x177777

## Process

# Assigned PCB Entry Format

Denne	0 1 2 3 4 5 5 7 8 9 10 11 12 13 14 15 15 18 18 18 18 18 18 18 19 18 19 18 18 18 18 18 18 18 18 18 18 18 18 18	
PCB01	SLL RELATIVE ADDRESS OF PROCESS' SEGMENT LOCALITY LIST	SLLPTR
PC802	8  /i	08X0SINF0
PC803		STKINFO
PC804		
PC805	FRTHER'S PCB INDEX	FATHERINFO
PC806		SONINFO
PC807	BROTHER'S PC8 INDEX	8ROTHERINFO
PC808	i 181 1011	PIINFONIMPPIN
PC809		PROCSTRTE
PC810		EVENTFLAGS
		LASTREFSHAPSEG
PC812	SUPPRABLE CODE SEGMENT	
PC813	1	QUESEINGINFO

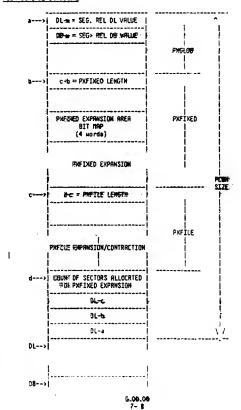
		Procees	Proces	•	
es i ana	ed PCB Entry Format (Cont.)			.(3:1)	MR. nail wait.
747.5	The Part & Change (Printer)	ì		(4:1)	BIG blocked I/O wast.
				(5:1)	TO, T/O wast.
8141	RIKTAX	I P8x		.(5:1)	UER UEOP wast and RIT wast.
٠	***************************************	1		.(7:1)	JNK, junk wait.
815		MRPDST		.(8:1)	TIR, timer wait.
1-	***************************************			. (9:1)	MSG. File system basic IPC message wait.
CB15	PIMP PCB INDEX	PIMPPIN		.(10:1)	SBX eon wait.
1-				.(11:1)	FR, father wast.
817	HIMP PCB INDEX	NIMPPIN		.(12:1)	Iff, procese waiting to be unimpeded.
1-				13:1)	SIR process wanting for a sir.
	BPTLIM	BPTLDM:		.(14:1)	TET, process wasting for a time out.
				. (15:1)	TOT process wasting for menory.
5191		HOPTR:	****		many many to many orders and an incident
820 I	PCB INDEX OF PREVIOUS PCB ENTRY IN QUEUE	POPTR	PC805	. (0:16)	FFM, Father's PCB relative index
	TED THOSE OF LUCY AND LES ENIM TO MODEL		PC806	. (0:16)	SARK son'e PCB relative index
			PC807	.(0:16)	992%; brother's PC8 relative index
800	.(D:1) SAR ==> scheduling attention required	, 1	PEROR	. (0:3)	PUB; pseudo - interrupt node
	.(1:1) Bounds Flag Privilege mode bounde	check		.,,	hard kill
	.(2:1) CRIT ==> procees is critical				2. soft kill
	.(3:1) HSIR ==> proceee hae e eir	1			3: stop
	.(4:1) PIOVR ==> pending PI, procese critica	al			4: hibernate
	.(5:1) MSPRI ==> hold eir priority	-			5. eecape
	.(6:1) IPEXP ==> incore protect expired				6: break
	.(?:1) PC ==> pre-engt canability				D: normal
	.(8:1) DSOFT ==> Delayed soft unt processin	s. A pendine		. (3:1)	ROFT, OH for soft interrupt to vake process
	soft int cannot be process	ed because of sur			even though it is waiting on another event.
	or critical state. PSEUDO	IMT will be invoked.		.(4:2)	DR+-
	when these condition(e) go				Q: other source
	.(9:1) LH ==> long wait	_			it. father
	.(10:1) SW ==> short walt				Z: 930
	.(11:1) TRW ==> terminal read wait				京: reply done on RIT wait
	.(12:1) USEDQ ==> ueed a quantum eince trane	action began		.(6:1)	DEMA, set during expiration.
	.(13:1) HIPRI ==> hold impeded priority			.(7:1)	FSC, of set, the father ie to be activated on proceed
	.(14:1) STOVA ==> proceeeing abort due to et				termotion.
	.(15:1) RITBK==> Request Information Table B	reak			1999
	44		PC909	(0:1)	LIVE set if process is alive.
1081	.(0:16) SLLPTR, SLL relative index to process	e' segnent		. (1:2)	BEE, block mail, valid if MA set
	locality liet				3. sent to father
2002	(Dat) 000 and if 00 accessor to the land	Add			7: received from father R: send to son
102	.(D:1) ROB, eet if DB pointing to an absolu				3: received eon
	.(2:14) XDS, DST entry number of extra data :	seducate to mutcu		. (3:2)	II received con  THE process to process communication, est with
	uo is eet; zero it none.			. (3:2)	respect to eon.
7967	.(0:1) STOVEREL FERS stack overflow is	Appendix addressed			at nuit
.003	.(1:2) SC, eet if executing system code	ori cont. Streets			% son to father
	.(2:14) DST entry number of process' stack				2. Father to son
	- (C. 14) has sucta unuses at biocess, struck				2 blocked
304	.(0:1) N, mourning wait.			.(5:1)	TW, etack overflow bit
	.(1:1) RG, global RIN wait.			.(6:3)	RYPE, procese type
	.(2:1) RL, local RIN wait.			- ( /	Œ uter
					6.00.00
	6.00.00				7- 6
	7- 5				· · ·

Proceee

Procees

PCBX Structure and Format

PEBX General Structure



ueer, eon of main user, main user, main, task eystem 1: ueer, eon of 2: user, main 3: user, main, t 4: eysten 5: 6: eyeten, UCOP 7: 7:
St. sat when the Dispatcher (and PSEUDODII)
enould be aware of a pending soft interrupt.
NH, hard hell pseudo interrupt
ST, soft hill pseudo interrupt
HB, hipernate pseudo interrupt
CY, control-y pseudo interrupt
BK, break pseudo interrupt
BK, break pseudo interrupt .(9:1) .(10:1) .(17:1) .(12:1) .(13:1) .(14:1) .(15:1) EVENTFLAGS, one for each wait class in PCB04 NS, wake up waiting switch set if an awake is missing. . (0:15) . (15:1) PC810 LASTREFSUMPSEG, segment identifier of last referenced swappable code segment. PCB11 . (0:32) (QUELING INFO)
DISPG => on dispatching queue
L scheduling class
C scheduling class
F scheduling class
F scheduling class
F scheduling class
INFEF ==> process is interactive
COMER ==> process is core resident
RSDFT, Allow seft interrupt. R value of 1
implies that user seft interrupt mill be
processed. R zero value inhibite user seft
inte (they are queued). This bit is namaged
by FINISTRIE and FINIENIT intrineice.
Process' echeduling priority .(0:1) .(1:1) .(2:1) .(3:1) .(4:1) .(5:1) .(6:1) .(7:1) .(8:8) PBM, CSTM block map index of procees' program. PCB14 .(0:16) MMPDST, DST entry number of the CST mapping table. PC815 .(0:16) PERFORM, PER-relative index of previous impades PER. PCB16. \_(0:16)-NUMPTING PCB" relative index of next inpeled PIN. PCR17 '. (9:16) PC818 .(0:16) BPTLINX, breakpoint link For procese PC819 . (0:16) MOPTR, PCB relative index of next proc in diep queue PQPTR,PCB relative index of prev proc in diep queue PC820 .(0:16)

Process

## PXGLOB Format

The PXGLDB portion of the pcbx is for job information, and contains the same job related information for all processes belonging to the same job.

2 3 4 5 6 7 8 9 10 11 12 13 14 15 --|--|--|--|--|--|--|--|--|--|--| OL-a=\$EG. AEL DL ∀ALUE OB-a=SEG. AEL DB VALUE USER ATTRIBUTES JMAT INDEX JPCNT INDEX JCUT INDEX 6 SB| A| TY | D| I|//|//|//| STACK OUMP FLAGS|6 7 //////////////// NATIVE LANGUAGE ACTUAL JOB INPUT LOEV ACTUAL JOB OUTPUT LDEV JOT DST INDEX 110 JIT DST INDEX 13

R = restart bit I = job in/list interactive D = job in/list duplicative TY = job type O = undefined 1 = session 2 = job 3 = task & = reserved:

SB= stun bit ;used for stack underflow simulation for ICF44 or ICF55.

Stack Ounp Flags
Bit 10 = Arned
Bit 11 = Suppress traceback
Bit 12 = Suppress ASCII
Bit 13 = Q-63 to S
Bit 14 = QINIT to S
Bit 15 = OL to QINIT

G.00.00 7- 9

Process

## PXFIXEO Assignments

The PXFIXED portion of the pobx contains specific information and control information.

		ı
0	c-b PXFIXEO SIZE	0
1	RELATIVE S(S-DB)	1
2		2
3 أ	INITIAL Q(Q-OB)	  3   LM MDST existed
4	INITIAL RELATIVE DL (DB-DL)	4 LP LOADPRDCed
5		1740 100es  5 .AT(0:1)-Arith.   17(1:1)-1:beams
6	AT  LT  ST  CY  CT  //  /	.CI(1:1)=Clorary  6 .SI(2:1)=System   .CY(3:1)=Ctl=Y
7	LINX TO XOS ENTRIES IN EXP. area   XOS CHT	7 .CT(4:1)-Code
10	P  S  EXTAR DATA SEGMENT DST INDEX	B L Logging   C Share Clock
11	P  S  EXTRA OATA SEGMENT OST INDEX	C Share Clock  9 G Global RIN acquired   A Acct UDC exist
12	I DI GI EUTOG AGTO GECMENT AGT THACU	IIO / O.I DECEDUEN EDD
13	P  S  EXTRA DATA SEGMENT OST INDEX	11   1:1 = 1 IF ABDAT
14	I XI AI ABOAT Y IAWI INITIAL CST INDEX	12 < 7:1 = 0 IF NAVE R/W
15		13   PADG FILE   1 DTHEAWISE
16		
17	ARITHMETIC TAAP PLABEL	15 \ AT PROCCREATION
20		16
21		17
		18
23	COOE TARP PLABEL	19
24	DATA CON TERMINATION TRAP PLAGEL	20
25	IMAGE TAAP PLABEL	21
26	RESERVED	22
27	CUA.MAX STACK SIZE(largest value ever for Z-DL)	23
	G.00.00	I
	7- 10	

Process

# PXFIXED Assignments (Cont.)

30	PAOCESS CPU TIME	   24
31	(MSEC)	25
32	MAXIMUM DATA SEG SIZE USED(IN SECTORS)	26
33	TOTAL VIATUAL STORAGE USED(IN SECTORS)	27
34	CURRENT EXTRA DATA SEGMENT SPACE	28
35	MAXIMUM EXTRA DATA SEGMENT SPACE	29
36	PRIV MODE BOUNDS FLAGS   STOV COUNT	30
37	PROCESS EXECUTION TIME REMAINDEA (IN MSEC)	31
40	SET TO-1 WHEN IN BREAK MODE*	32
41	CONTINUE FLAG (:CONTINUE COMMAND)**	33
42	ACTUAL SIZE OF VIATUAL SPACE ALLOCATED TO STACK	34
43	EAROR LEVEL	35
44	INTRINSIC ERRORS	36
45	INTAINSIC ERRORS	37
46	INTRINSIC ERRORS	38
47	INTRINSIC ERRORS	39
50	INTAINSIC ERADAS	40
51	INTAINSIC ERADRS	41
52	TSLR, virtual time since last rescheduled	42
53	ISIB, virtual time since transaction began	43
54	TSSWAPIN, virtual time since swapin	44
55	TSLA, virtual time since last absence	45
56	TSLD, virtual time since last deallocation	  46
57	QCNT, quantums used since transaction began	47

PXFIXED Assignments (Cont.)

60 | / | D | / | D | AESERVED FOR FUTURE SOFT INT USE 148 |/|C|/|S| |/|Y|/|I| 61 TRLX INDEX FOR KEANEL TIMEOUT PROCEDURE 49 JOB TYPE: 50 1=SESSION 2=JOB JOB/SESSION NUMBER 63 <--- (reserved )-----51 RESERVEO FOA FUTUAE USE 52 64 RESERVED FOR FUTURE USE 53 AESERVED FOR FUTURE USE 54 66 67 RESERVED FOR FUTURE USE 55 70 56 |CY| |SI| 71 TIMEOUT TRIX 57 58 59 74 PCLASSMASK 60 75 PROCQUESTOPHDA0 61 76 62 PROCSTOPTIME 77 63 UNUSED 114 PXFIXED EXPANSION BITMAP 117

NOTES: P = 1 if opened by priv user S = 1 if data segment is sharable

PCLASSINSK = BIT MASK OF CLASSES THIS PROCESS HAS EMBLED PROCQUESTOPHORD.(0:4) = PROCESS PRIORITY: 7 => L QUEUE 6 => C QUEUE 2 => D QUEUE 1 => E QUEUE

G.00.00 7- 12

6.00.00 7-11

.(4:12)= REASON STOPPED: 1 => STOP SEG FRULT
2 => STOP DISC URIT
3 => BLOCKED I/D, NON TERMINAL
4 => TERMINAL READ
5 => STOP IMPEDE
6 => STOP PROTISE
PROCSTOPTIME = DBL WORD TIMESTAMP OF WHEN PROTESS STOPPED FOR
REASON GIVEN IN PROCQUESTOPWORD

Process

DCY

A DELAYED CONTROL Y IS PENDING (TMIS BIT IS CHECKED BY JAIN ON BOUNDS VIDITION TO OFTERRINE IF GOT: 1) TRUE BOUNDS VIDITION OR 2) RN INDUCED BOUNDS VID THAT INDICATES THAT THE CONTROL Y TARP PROCEDURE MRY NOW BE ENTERED).

DSI STATE OF THE "RSOFT" PCB BIT WHEN CONTROL Y TARP WAS ENTERED. RSOFT = 1 RILLUS USER SOFT INTERCUPTS RGAINST THE PROCESS. IT IS SET TO ZERO WHEN THE CONTROL Y HANDLER IS ENTERD.

IT IS SET TO ITS PRIOR STATE WHEN THE USER CALLS RESETED THE CONTROL Y HANDLER IS ENTERD.

\* SET TO COMMAND RECORD LENGTH WHEN COMMAND PENDING (1.E. COMMAND RECORD LENGTH WHEN COMMAND PENDING (1.E. COMMAND ENTERED OURING BREAK OR ENCOUNTERED DURING FLUSHING).

\*\* CONTINUE FLAG VALUES

0 = NO CONTINUE IN EFFECT

1 = CONTINUE JUST ENCOUNTERED

2 = CONTINUE IN EFFECT FOR THIS COMMRND

PCBXFIXED(56).(1:1) = SET BY PSEUDDINT WHEN THERE IS R PENDING CONTROL Y WHICH CANNOT BE PROCESSED BECRUSE OF SYSTEM CODE OR PRIVILEGEO CODE. ININ CHECKS THIS BIT ON BDUMDS VIDLATION DR TRRCE TRAP.

SI FLAG

PCBXFIXEO(56).(3:1) = SPECIFIES THE STRTE OF THE USER INTERRUPT FLAG WHEN THE CURRENT CONTROL Y WRS PROCESSED.

## PXFIXED Expansion Bitmap

G.00.00 7- 14

## Process

## PCBX For Core Resident System Process Stacks

٥	DL-a (Seq Rel DL Value)	0
-1	DB-a (Seq Rel DB Value)	1
2	USER ATTRIBUTES (always -1)	2
3	0	3
4	0	PXGLD8  4
5	0	5
6	0   0   1   0	6
7	O	<sub>7</sub>
10	RCTUAL JOB INPUT LDEV	8
11	ACTUAL JOB OUTPUT LDEV	9
12	0	10
13	0	<sub>11</sub>
12	PXFIXED SIZE (c-b)	10
13	RELATIVE S (S-DB)	11
14	AELATIVE Z (Z-D8)	12
15	INITIAL Q (Q-DB)	13
16	RELRTIVE DL (DB-DL)	14 PXFIXED
17	GENERRL RESOURCE CRPRBILITY(-1)	15
20	RESERVED	16
21	0	17
22		18
23	DL-b	19
24	DL-a	20

NOTES: 1. There is no PXFILE area.
2. The PXFIXED area is much smaller than a normal PCBX.

Process

## Process To Process Communication Table

This table is used as the communication link by which father and son processes communicate with one another via the wallbox scheme. This table contains two words per entry and is indexed by PCBM (entry index 0 is meaningless). Each two word entry of index N essentially relates where, as well as how much, wall may be found for a process N with respect to communications between N and his father process.

ENTRY FORMET

word 0 WORD COUNT MAIL WORD OR DSTN Hord 1

where word 0 = the # of nail words to be transferred. word 1 = the only word of mail itself if word 0 = 1 itself if word 0 = 1 otherwise it contains the DST# of the extra data segment where "word count" words of mail exist.

NDTE: Assume process S is the son of process F. Then the process to process communication table index which will be used for mailbox communication between son S and father F will be that of the son (i.e. S).

## Subsystem Reserved DL Rrea

|-----

## REMAINING DL ARER

DB-12	RESERVED FDR SDRT/MERGE	DB-10
DB-11	RESERVED FOR TRRCE, TODLBDX, & BUSINESS BRSIC	DB-9
DB-10	EXTERNAL PLABEL OF DUTER BLDCK	D8-8
DB-7	RESERVED FOR TRRCE & SYMBOLIC DEBUG	DB-7
DB-6	DB RDDRESS DF STLT	D8-6
DB-5	RESERVED FOR CDBDL	DB-5
DB-4	RESERVED FOR COBOL	DB-4
DB-3	RESERVED FOR COBDL	DB-3
DB-2	RESERVED FOR FORMATTER & PRSCRL	DB-2
DB-1	DB RDDRESS DF FLUT	DB-1
- 1		1

DB RRER

## FDRTRRN Logical Unit Table (FLUT)

The segmenter is responsible for the preparation and initialization of a FDRTRAN logical unit table. This is done when a program is prepared if that program contains at least one program unit that references a logical unit. The location of the FLUT is in the secondary DB area and the address of this location is contained in DB-1.

The FLUT is formatted as per the following example:

D8-1	   x 
DB+X	13101
אייטע	1 1 1 1
	!!
	4 0
	i 5 i 0 i
	iii
	17101
	10   0
	j255 j/// j
	II
	1

2nd BYTE
The MPE file number (as returned by FDPEN) used in accessing the file. Zero if file not open.
Filled in by formatter as each l.u. is initially referenced.

								9	10	11	12	13		15	
	 1		1		1	1	1	- <del>-</del>			I	I	<del>-</del> -	I	
j	 								<b></b> -						į
ļ- <b>-</b> -	 														i
ĺ															Ĺ

G.00.00 7- 17 G.00.00 7- 18

### CHAPTER 8 JDB TABLES

## Job Tables Overview

- Job Master Table (JMRT): One entry per job/session. Contains information needed to get the job/session running. Entry is created at the introduction of job/session.
- Job Information Table (JIT): Dne DST per job/session. Contains information needed by the job/session as it is executing.
- Job Process Count Table (JPCHT): One entry per job/session. Entry number used to index into the JIR to lock job resources.
- Job Directory Table (JDT): Dne DST per job/session. Contains the following sub-tables used by descendants of job/session. Must obtain JIA (by using JPCNT index) before accessing JDT. Sub-tables:

  1. Data Segment Directory Directory of DSTs used by job/session

  2. Temporary File Directory

  3. File Equation Table

  4. Line Equation Table

  5. Job Control Word Table
- Job Cut-off Table (JCUT): Stores total CPU time limit of job/session and accumulates the CPU time that job/session uses.
- Ucop Request Queue: A queue of Process Identification Numbers that are terminating.

G.00.00 8- 1

Job Master Table Structure (JMAT)

SIA = 15(10) = X17 DST = 25(10) = X31 2ERDTN ENTRY D 1 2 3 4 5 6 7 8 9101112131415 max JMRT size (words/128) current JMRT size (words/128) :VMOUNT state saved for WRRMSTRRTs JMRT entry size (38) DB pointer to first entry (38) MAXSIZE | CHASTZE VMOUNT INFO | ENTRY SIZE ENTRY POINTER DB pointer to word 0 of head SCHEDULING HEAD POINTER entry in scheduling queue
DB pointer to word O of tail
entry in scheduling queue
next assignable session #, TY=1 SCHEDULING TAIL POINTER SCOUNTER TY TY JCOUHTER next assignable batch #, TY=2 LG=1, logoff in progress SEC=0, high;=3,low JOBSECURITY maximum number sessions C E 11 LG SEC |//////SFENCE/JOBFNCE 9 12 SLINIT 1D SHUM 13 11 current number sessions JLIMIT 12 naximum # batch jobs JNUM 13 current # batch jobs JMAT SCHEDHERD 14 16 **UDAKAREA** 15 SFENCE is session fence 17 (23UDS) 20 451 38 46

6.00.00 8- 2

Job Tables

JNRT (Cont.) ENTRY 1 75 113 LRST ENTAY SCHEDULING QUEUE HATTING SESSIONS
FIFO WITHIN NIPRI/INPUT PRIDRITY
[ERROR JOBS ]
[ FIFU ]
WRITING JOBS

FIFD WITHIH HIPRI/IMPUT PRIDAITY

Job Tables

## Job Master Table Entry (JMAT)

1 1 1 1 1 1 0 0 0 1:2:3 4:5:6 7:8:9 0:1:2 3:4:5 | D state | D = free entry | 1 1 = introduced, in | STRATDEVICE | 2 %70 =scheduled in scheduled job queue. state :0|I:G:A|U:C: INPRI jab/session number ty: iob/session # 240 = waiting, job in scheduling queue 260 = initial, UCOP has created JSMP finished initial. 3 = terminating. 4 = suspended. 10 account mane = suspended. = duplicative = interactive 10 13 iob name 113 20 21 22 group logon name 116 18 C = JLIST is device class index 23 19 JIN device 20 24 JLIST device 25 Julian date (CALEHDAA) 21 time (CLOCK) 22 23 26 27 30 j language XPRI 24 31 25 Main pin CPU lim. (O deflt, -1 no lim.) 26 321 33|S|R:H:FT : OUTPRI : NUMCOPIES i 27 DAIGJIH/DAIGJLIST is used as a scheduling link by UCDP (state= %40). DB relative ptr. Last entry in list contains zero (0) 34 ORIGJIN 28 29 DRIGHTST 35 36 Aeserved 130

JOB STATES - JMAT ENTRY WORD O. (0:6)

LOGON USES ALL STATES EXCEPT "SUSPEND"

SMOWJOB - Displays job states by scanning JMRT DST (X31)

## JMAT (Cont.)

		1
37	Aeserved	31
40	Reserved	32
41	Reserved	33
42	Reserved	34
43	Reserved	35
44	Unused	36
45	Unused	37

0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5 1 1 1 1 1 1

R = RESTART N = SEQUENCED S = ORIGJIN is spooled.

FT = funny terminal
00 - regular term.
01 - regular term.
special logon
10 - RPL term.
11 - RPL term.

6.00.00 8- 5

STATE | STATE NAME PROCESS | SEGMENT | PROCEDURE(S) NO. STARTOEVICE ->PUTJMAT ->ALLOCENTRY IN SEGMENT ALLOCUTIL INTRO DEVREC NURSERY JSMP SPOOLER UCOP SCHED CXSTSTREAM 270 IJOBSCNED SCHEDULEDSCHED DEVREC JSMP **740** MAIT NURSERY STARTDEVICE ->SCHEDULEJOB SPOOLING SPOOLSTUFFIN ->SCNEDULEJOB SPOOLER 1/ INIT-IRLIZRT-ION X60 I UCOP I UCOP LAUNCHJ08 2 EXEC 1 JSMP | NURSERY | INITJSMP TERMIN-ATING TERMINATE ->EXPIRE -> CLEANUPJOB 3 MORQUE TERMINATE ->EXPIRE ->
CLEANUPJOB ->DEALLOCENTRY | FREE Ð JSMP MORQUE

Job States

For states INTRO and WAIT,

ENTAY

| SUSP

DEVREC => logon command originated on terminal or other unspooled device.

SPOOLER => logon command originated on spooled device.

JSMP => logon command is the result of the execution of a :STRERM command. (This also includes USER processes which have done programmatic :STRERMs.)

| OPLON

G.00.00 8- 6

IN ALLOCUTIL

| CXBREAKJOB

Job Tables

Job Process Count Table (JPCNT)

(1 Bit Entry/Running Job )

MEMORY RESIDENT

SYSGLOB BASE = DB+13(X15) DST = 24(10) SIA = 13(10)

01234567891012345 Total Configured number of Jobs and Sessions Total number of free entries Bit Map relative index of word containing next free entry 3 unused Bit Map 64 words long

free entry = 1 allocated entry = 0

A JPCNT entry must be allocated before the nain process can be procreated.

The job SIR (PXGJSIR) = some base+JPCNT index.

NOTE: This table is completely bit oriented with each entry consisting of one bit. Entries are taken from available pool on a "first found" basis. R "!" found in the bit map indicates a free entry. A zero (0) found in the bit map indicates an allocated entry. Nord 2 of this table is the index of the word in the Bit Map where the next free entry resides. At system start up, this word is set to zero (0). The Bit Map can be thought of as ranging from 0-63 (64 total words - 1024 entries).

Job Tables

Job Cutoff Table (JCUT)
1 Entry/ CPU-limited Job

MEMORY RESIDENT

SYSGLOB BRSE = DB+11(X13)
DST = 36(10);SIR = 14(10)
SYSGLOB + X117 = default
CPU tine limit for jobs

		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
			0
		ENTRY SIZE (3)	1 NEADER
		FAEE NEAD	ENTRIES 2
¦ -		POINTER TO LAST ENTAY (O)	(2)  3
Н		UNUSED	4
		UNUSEO	5
			TYPICAL ENTRY
ij		JCUTCPUL	time limit (seconds)
i i		JCUTCPUC	tine count (nsec)
     	İ		
i i 	->	PDINTER TO MEXT FREE ENTRY (END OF LIST = D)	
į			FREE ENTRY
i			
-	->	LRST ENTRY	

G.00.00 8- 7

# Job Information Table (JII) JIT OST 18 word 11 (base 10) in PXGLOB

(	1 1 1 1 1 1 0\1:2:3\4:5:6\7:8:9\0:1:2\3:4:5		
٥į	JIT DST	0	
ᅦ	6 ; not used	1	
2	pointer to job info 8	2	
3	pointer to acct info 48	3	
4	pointer to recerved area 59	4	
5	aeeociation table index	5	
6	F	6	F - Job/Session-wide FPMRP option flag
10	ty : job number	7	(JSFPNAP) ty - 1 = Seseion 2 = Job
11	7	9	2 - 344
12	JITMRXP :EOF:	10	JITHRXP - MRXJOBPRI capability JITHPN - Job main PIN.
13	JITMPN	11	<pre>JITEOF - used by FCLOSE to tell CI that a \$STDIN(X) file was closed</pre>
14	DS ORTRSEG	12	<pre>#/out encountering an EOF. (0:1)=\$STOIN, (1:1)=\$STOINX</pre>
15		13	<b></b> ,
16		14	
20		16	
24	JITHGN (4 words) hane group	20	
30		24	

JIT (Cont.)

1 1 1 1 1 1 0i1:2:3|4:5:6|7:8:9|0:1:2|3:4:5 28 |29 |30 |31 34 | 35 | 36 | 37 | JITUN user name 53 | 32 40 pointer to JITRIP P - Group's home volume is a private volume n - Private volume nounted (i.e. group bound to home volume set), JITGIP = 57 41 PIM: pointer to JITGIP
42 LATIR 55 33 34 local attributes PRSSF passed file pointer 451 38 UCAP user capability \* 50 Reserved for DS'II 40 43 53 local RIN pointer 44 45 46 47 54 55 56 57 JITJN 0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5

G.∞.00 8- 9

5.00.00 8- 10

## Job Tables

## JIT (Cont.)

```
1 1 1 1 1 1 0 0 1:2:3 4:5:6 7:8:9 0:1:2 3:4:5
                                              Accounting Info
60
61
    JITCREC - # of creations
                                         149
            JITCPUC
cpu nilliseconde
                                         |50
|51
62 i
63 i
                          NIPRI
                                         52
                                                 NIPRI - highest job priority
       not used :
                                        |53 Account
|54 Index Pointer
65
66
                  JITĂIP
                                         | SS Group index pointer
| SS System volume set
67
70
                  1IIĞIP
                                         71
72
                           HVTRBX
                  JITGIP
 73
                                      0 160
 74
75!
76!
77!
100!
101!
102!
                                         61
62
63
64
65
66
           allou mask
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
```

## Rilow Hask Format

The Rilou mask for MPE V is expanded to eix words. There is a mask in each user's JIT and in the SYSGLOB area. The Rilou mask contains enough bite for a one-to-one correspondence to every present OPERRIOR type command, or any future OPERRIOR command. When a user is RILOUSED and command or RESOLIRIED to a device (which will use OPERRIOR type commands) then the corresponding bit(s) in the mask in that user's JIT for that command is set. If the RILOUS or RESOLIRIE was done on a global scale, then the bit(s) in the mask of the SYSGLOB area is/are updated.

The Following EQUATEs define the mask bit for each operator command.

The first set of commande define the operator commands dealing with

## Job Tablee

When adding a new command to this set of EQUATEs, be sure to add a corresponding move statement in LOGITAGE, even if the command will not be

<u> Word Bit</u> ₩

REORTIO SCIEPT JOHN GIVE GIVE HERDORF HERDON REFUSE REPLY STRATSPOOL TAKE UP MPLIME DSCONTROL UPPER LIMIT->0EV	O O O O O O O O O O O O O O O O O O O	0 · 2 3 4 5 6 7 8 9 10 11 12 mans	0 1 2 3 4 5 6 7 8 9 10 11 12
RORTIJB RALLOW RALTILE RALTIJB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB BREAKJOB LIMIT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LOUNT LO	0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	13 145 0 1 2 3 4 5 6 7 8 9 10 11 2 3 4 5 6 7 8 9 10	13 14 15 16 17 18 19 20 1 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42

FOREIGH 2 11 43 INF 2 12 44 SHOWCOR 2 13 45 OPENQ 2 14 46 SHUTQ 2 14 47 GISCAPS 3 2 47

A THE FORMAT FOR UCRP (246-47) IS RS FOLLOWS:

	0  1  2	31 41 51	61 71 81	13 14 15
MORO1	SNIANIALIO			
HORDS			8A IA F	

Job Directory Table (JDT)

0	MRX SEG SIZE(WOS)	1 entry per job DST W in word 10		
1	POINTER TO JOSD	(base 10) of PXGLOB		
2	POINTER TO JTFO			
3	POINTER TO JFEQ			
4	POINTER TO JLEQ			
5	POINTER TO JJCN			
6	POINTER TO FREE SPACE			
	NORK ARER 15 words			
JOSJNUM	TY NUM	job number		
	JSMPIN	main process number		
J050	JOB DATA SEGMENT DIRECTORY			
JTF0	JOB TEMPORARY FILE CIRECTORY	  EHTAY   NAME   SIZE (HDS)  SIZE (HDS)		
	    	C1   C2		
JFEQ	JOB FILE EQUATION TABLE	CH ( (240)		
JLEQ	JOB LINE EQUATION TABLE	ENTAY INFORMATION		
	JOB CONTROL HORD TRBLE (JJCH)	The name is a		
	FAEE SPACE	concatenation of up to 3 subnames. Bit 0 of the 1st character of each subname is 1.		

G.00.00 8- 14

G.00.00 8- 13

Job Tables

Job Data Segment Directory Entry (In JDT)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 			
SEGMENT IO			
EXTRA DATA SEGMENT DST INDEX			
W OF PROCESSES ACCESSING			

NOTE: A return of X2004 in the INDEX value after using the GETDSEG intrinsic indicates that there is no nore room in the Job Directory Table for another job sharable data segment.

Job Temporary File Entry (In JOT)

NAME-ACTUME FILE DESIGNATOR	concatenation of up
VOLUME POINTER   FILE LABEL POINTER	to three subnamee.  81t 0 of the first character of each subname ie 1.

Job Tablee

File Equation Table Entry (In JDT)

NAME LENGTH (BYTES) | DEVICE LENGTH (BYTES)

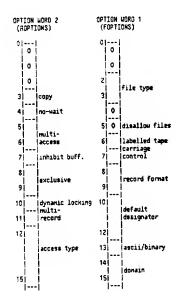
MANE-ACTUAL DESIGNATOR (may not be present)

|-----|

Job Tables

Roptions and Footions Word Breakdown

6.00.00 8- 17



Job Tables

----- PMRSK WORD 2 IO BLOCK FACTOR FILE TYPE RECSIZE LABELLEO TAPE FAMS MESSAGE DISPOSITION NUMBUFFERS USER LABELS INHIBIT BUFFERING LANG VTERM EXCLUSIVE MULTI-RECORD POINTER ENTRY ACCESS TYPE DYN. LOCKING WAIT, NOWAIT COPY, NOCOPY CRRRIAGE CONTROL MULTI RECESS RECORD FORMET NUNCOP OUTPRI DEFRULT DESIGNATOR ASCII/BINARY FTLECODE DONAIN FILESIZE NUMERTS DEVICE NAME INIT ALLOC

> 1->info present 0->info absent

G.00.00 8- 18

PMRSK Word Breakdown

OI HAMM REQ EMTRIES  1 TRBLE RELATIVE POINTER TO N  2 TRBLE RELATIVE POINTER TO  3]  O	N/2 EXT GVAIL ENTRY  NEXT REQUEST	UCOP Entry Format   Each entry is 2 words icing
REQ 1  REQ 2		
	5.00,00 8- 21	5.00.00 8- 22

#### CNAPTER 9 RELOCATABLE OBJECT CODE

#### USL Files Introduction

\* USL record length 128 words always. \* Layout of doubleword disc addresses



- $\mbox{\scriptsize {\star}}$  Hash links join all entries with the same hash key regardless of
- \* hash links join all entries with the same mash key regardless of type.

  \* Linear lists terminate with a zero link

  \* Circular lists containing only the list head point directly to themselves.

  \* Single-word disc addresses

	9-BIT	RECORO	#			NOF HITNIN	D # Record	
0				8	9			15

Uninitialized fields are reserved for future use and should be set to zero.

#### Record O and Overall USL File Format

					NOTE:	
0	LIO	0	LOADER ID	S.A.	= Starting	Address
1	NE	1	NR. DIRECTORY ENTRIES			
2	0 L	2	DIR. LENGTH			
3	SUMDG	3	TOTAL DIR. GARBAGE			
4	NDG	4	NR. DIR. GARB. ENTRIES			
5	SRBDL	5	S.A. BLOCK DATA LIST			
6	SAIPL	6	S.A. INTERRUPT PROC. L	IST		
7	\$AS L	7	S.A. SEGMENT LIST			
10 11		8	FILE LENGTH			

G.00.00 9- 1

#### Relocatable Object Code

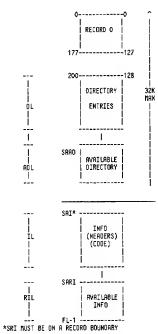
#### USL File Format (cont.)

12	SARD	10	S.R. RVRIL. DIR.
13	RDL	11	AVRIL. DIR. LENGTH
14 15		12 13	S.A. INFO BLOCK
16 17	īL	14 15	INFO BLOCK LENGTH
20   21	SARI	16 17	S.R. AVAIL, INFO
22   23	AIL	18 19	AVRIL. INFO LENGTH
24 25	TOTRL	20 21	TOTAL INFO GARBRGE
26	NIG	22	NR. INFO GARB. ENTRIES
27		23	
30			
31		   25 	
32		26	
33		27	
34		28	
35		29	
36		30	
37		31	
40		32	
41	HL 0	33	NASH LINKS
	:		
177		127	7

6.00.00 9- 2

# Rélocatable Object Code

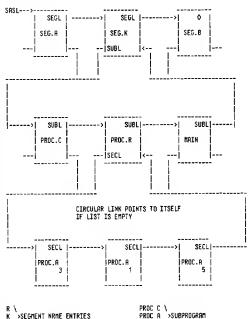
# USL Files General Information (cont.)



NOTE: ALL RODRESSES IN RECORD O ARE MORO ADDRESSES.

#### Relocatable Object Code

#### USL Files General Information (cont.)



R \
K >SEGMENT NRME ENTRIES
B /

PROC C \
PROC A >SUBPROGRAM
MAIN / ENTRIES

G.∞.∞ 9- 4

> SECONDRRY ENTRY POINT ENTRIES

#### Data Descriptors, Passed Parameters

0123456789	
- - - - - - - - -	
MODE   STRUCTURE	TYPE

TYPE	HDADS	CODE
NULL LDGICAL INTEGER BYTE REAL OUBLE LONG COMPLEX LABEL (SPL) CHRRCTER (STRING) LBEL (FORTRAN) UNIVERSAL (HRTCHES RNY TYPE)	1 1/2 2 2 2 3 4 N/2	0 1 2 3 4 5 6 7 10 11 12
STRUCTUAE SIMPLE VRRIABLE POINTEA RRRRY PROCEDURE		0 1 2 3
HOOE		
NULL VALUE REFERENCE MRME		0 1 2 3

NOTE: A descriptor of O results in an automatic match.

#### Pascal

Pascal sets the high order bit in the parameter type descriptor when it is generating hashed values. The remaining 15 bits are based on a hash of the types of the parameter. Only the Pascal compiler can compute the value, and the SEGMENIER must match the whole 16 bit value.

G.00.00 9-5

Aelocatable Object Code

Clarification Notes on Entry Types 2 and 4 With Respect to SPL and FORTARN

*ENTRY TYPE 2 SPL D.B.	**ENTRY TYPE 4 SPL PRDC	*ENTAY TYPE 2 FORTRAN MAIN	**ENTRY TYPE 4 FOATARN SUB.
TPOB	0	0	0
1,5 TSDÐ	1 1808	1,2,3,4 TSDB	1,2,3,4 TSDB
NUPUST	NUPUST	NUPUST	NUPUST
5 NHSDB	HUO	NUD	KHO

HHERE: TPDB = Total primary DB length in words
TSDB = Total secondary DB length in words
NUPUST = Humber of words in "TRRCE" array
HUSDB = Number of words in secondary DB array
HUD = Number of words in own array
HUD = Number of words in data array

Hotes:

Does not include the length of the SILT
 Does not include the length of the FLUT
 Does not include the length of any common array
 Includes the length of any OB-allocated format array
 Re not necessarily equal

In general TPDB and TSDB are summations of storage allocated in the global area of the program's data segment. They are not, however, complete since the compilers are not aware of all storage actually allocated! The STLT and FLUT are examples of this since these tables are constructed by the segmenter. Common arrays also present a problem since their inclusion in TPDB and TSDB might cause their storage requirements to be counted more than once.

Relocatable Object Code

#### Entry Type 0

#### GRABRGE

0 1		1D 1	1	15	
1///1	NU		0	i	NW - Number of words in this block
	GARBRGE				

# Entry Type 1

# SEGMENT NRME

0	1	7 8	10 11	
//		NU	1	K
1		H L		I H
IR I	///////	NC	CHAR1	1
-	(VRRIABLE	# CMBO	SEE NU)	į R
į	(**************************************		occ no,	Ì
1	CHAA. NE	1////	/////////	: ///  N
ī		SEGL		
1 1		SUDI		

- NW Number of words in entry block
- HL Hash link points to next entry having the same hash code
- Retivity bit 0 if active 1 if inactive (initialize to 0)
- Note: An inactive segment implies that all entry points are inactive
- | L | SUBL | HC Munber of characters in name. Max is 16

CHAR. 1 - First character in

CHMR. 1 - first character in variable field
CHRR. NC - Last character in variable field
SEGL - Segment link - points to next segment name

next segment name entry
SUBL - Subprogram link - points
to next entry having
the same segment name
L - Last entry in list
O if not last
1 if last

G.00.00 9- 6

Relocatable Object Code

#### Entry Type 2

OUTER O	BLOCK 1 2 3 4 5 6 7 8	10 11	15
1//1	HH	2	١
1	NL		Ī
A	C   I  ///  HC	CHRR 1	1
	(VAAIABLE # CHAA.S	EE HC)	
I	CHAR NC  /////	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	771
L	SUBL	<b></b>	
1 4 1	SECL		1
l	SSR		1
	SRC AELATIVE TO SRI (SEE I	RECOAO D)	1
F	N   NNC		Ī
l	SE		Ī
1	TPDB		Ī
1	TSDB		Ī
l	HUPUST		Ī
1	NHD/NHSDB		Ī
T	NH		Ī
	SRH RELATIVE TO SAI (SEE	RECORD 0)	
	HDU		Ī

#### Entry Type 2 (cont.)

1	. !
	•
	<u>-</u> <u>-</u>
1	HDM
į.	. !
	. !
	• !
T	NH ]
	SRH
İ	i
	HDH
1	.
i	. 1
I	. 1
	HDH

- NW Number of words in entry block.
- HL Hash link points to next entry with same hash code.
- A Activity bit. O if active, 1 if inactive outer block.
- C Callability bit set if entry point is uncallable.
- I Privilege node bit set if program unit is to be executed in Privilege node..
- NC Number of characters in name. Max is 16.
- CHAR. 1 First character in variable field.
- CHRR. NC Last character in variable field.
- L Last entry in līst. O if not last 1 if last

G.∞0.∞ 9- 9

#### Relocatable Object Code

#### Entry Type 2 (cont.)

- SUBL Subprogram link points to next entry Entry having the same segment name.
- SECL Secondary entry point list link.
- SSR Program unit starting PB address.
- SRC Starting 8FILE9 address of code module
- F Set if fatal error
- W Set if nonfatal error
- NWC Number of words in code module.
- SE Stack size estimate
- TPDB Total number of words of primary DB to be allocated
- TSD8 Total number of words of secondary D8 to be allocated.
- NUPUST Number of words in trace array (PUST)
- NWD Number of words in data array (FDRTRRN)
- NWSDB Number of words in secondary DB array (SPL)
- T Terminating bit set if last set of headers in entry
- NN Number of headers
- SAH Starting address of header (relative to SAI)
- HDW Header (pointer)

G.∞.∞ 9- 10

# Re'locatable Object Code

# Entry Type 3

#### DUTER BLOCK - SECONDARY ENTRY POINT

0	1	2	3	4	5	6	7		В		0	11		15
1//					H	4					ı		3	
1						Н	L							¦
A	1 8	17.	/1/	71	ł	NC		Ī		CI	IAR	1		1
	(VARIABLE # CHAR. SEE NC)													
Ī	C	HAR	NC				1/	11	111	////	1//	"	///	///
L	I				:	SE	C.L							
1						SSI	9							1

#### Entry Type 4

#### PROCEDURE

0 1 2	3 4567 8		10	11	15
1//	ни				4
1		HL			
H   C  I	H  NC	CHA	R.1		
	(VARIABLE	# CHRR.	SEE NC	)	
CHRR.NC	1///	111111111	//////	[][][]	//////
IL į	SUBL				
L	SECL				
1	SSA				

#### Relocatable Object Code

# Entry Type 4 (cont.)

	SAC
F   W	NuC I
1	SE I
	TPDB I
1	TSDB
I	NWPUST
1	NUD/NUO I
P	NP   CN
1	TH I
1	PARM.1
	(VRRIABLE # OF PRRMS. SEE CM)
1	PARM. NP
1	NN
	SAH
I	HDH [
	:
ı	HDW I
	:
I	ETC

#### Reincatable Object Inde

#### Entry Type 4 (cont.)

NW - Number of words in entry block
NL - Nash link - points to next entry with same hash code
R - Retivity bit 0 if active, 1 if inactive entry point
C - Callability bit set if entry point is uncallable
I - Privilege node bit. Set if procedure is to be executed in privilege node.
N - Nidden entry point. Set if entry point will not be in
library directory.
KC - Number of characters in name. Max is 16.
CHARI - First character in variable field.
CHARI - First character in variable field.
L - Last entry in list
0 if not last
1 if last
SUBL - Subprogram link. Points to next entry having the same segment
Name
SECL - Secondary entry point list link.
SSA - Unit starting PB address
SRC - Starting (file) address of code module
F - Set if fatal error
U - Set if fatal error
UNC - Number of words in code nodule
SE - Stack size estimate
TPDB - Total number of words of primary DB to be allocated.
TSDB - Total number of words of secondary DB to be allocated.
MAPUST - Number of words in data array (FORTHARM)
NNO - Number of words in data array (FORTHARM)
NNO - Number of words in data array (SPL)
P - Parameter checker
On on checking. (Inplies NP undefined, FN and PRRM's absent)
01 check procedure type and number of PRRM's (inplies PRRM's absent)
10 check procedure type, number of PRRM's and type of each PRRM's
absent) absent)
10 check procedure type and number of PARM's (implies PARM's absent)
11 check procedure type, number of PARM's and type of each PARM.
NP - Number of PARM's
CN - Character count of PARM's
TN - Terminating bit. Set if last set of headers in entry.
NM - Number of headers
SRH - Starting address of header
NOW - Neader (pointer)

G.00.00 9- 13

Relocatable Object Code

#### Entry Type 6 INTERAUPT PAGCEGUAE

0 1  2  :	3  4567 8	10 11 15
1//1	NU	6
	NL	
[A   IT	//  NC	CHRR.1
(VAI	AIABLE # CNAR	. SEE NC)
A   IT	//  NC	CNAA.1
(VAI	AIABLE # CHAA	. SEE NC)
CHAR. NO	1////	(11111111111111111111111111111111111111
I	IPL	
I	OBS	1
1	SSA	·····
	SAC	     
F   W	NHC	ı
<b>!</b> T	HN	I
	SAH	
1	HDU	I
	:	) 
1	HDN	1

Relocatable Object Code

#### Entry Type 5

PACCEDURE - SECONDARY ENTRY POINT

0 1 2 3	4567	8 10	11 15
//	NU		5
		HL	
A] C  // N	I NC	CNAR	. 1
(VARIABLE	#CNAA. S	EE NC)	
CHAA.	NC	1//////	////////
ļ. i	SECL		
	SSA		
1		- <del></del>	

NW - Number of words in entry block

NL - Wash link - points to next entry with same hash code

A - Activity bit. O if active, 1 if imactive entry point

C - Callability bit set if entry point is uncallable.

H - Hidden entry point set if entry point will not be in library directory

NC - number of characters in name, max is 16

CNAA 1 - First character in variable field.

i - Last entry in list 0 if not last 1 if last

SECL - Secondary entry point list link

SSA - Unit starting PB' address

G.00.00 9- 14

Relocatable Object Code

#### Emtry Type 6 (cont.)

NW - Number of words in entry block

NL - Nash link. Points to next entry with same hash code

A - Activity bit. O if active, 1 if inactive entry.

IT - Interrupt procedure type number

NC - Number of characters in name (maximum is 16)

CNAR 1 - First character in variable field.

CNAA NC- Last Character in variable field

IPL - Interrupt procedure link

DBS -Number of words of OB storage required.

SSA - Unit starting PB' address

SAC - Starting (file) address of code module.

F - Set if fatal error

W - Set if nonfatal error

NHC - Number of words in code module

Terminating bit. Set if last set of headers in entry.

NN -Number of headers

SRH - Starting address of header.

NDW - Meader (pointer)

#### Entry Type 7

BLOCK DATE

0   1   2   3  4	567 8	10   11	15
	NH 		7
	HL		
8   F   W  ///	NC (	CHAA.1	
BFOCK	DATR NA	ME 	
CHRA.NC	1////	11111111111111	//////
	BDL		1
1	CAL		1
1//////////////////////////////////////	NE	CHAR.1	ı
COMMO	N AARAY	NAME	    
CHAR.NC	1////	11111111111111	//////
T	NH		I
	SAN		
1	HDU		1
	:		   
	NDW		!

G.00.00 9- 17

#### Relocatable Object Code

Entry Type 7 (cont.)

1		CAL		
1////	11/1/////	NC	CNRR.1	1
	COMMO	IN RRRRY	NAME	
1	CNAR.NC	1///	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	/////
1		NH		I
		SRH		
1		HDN		I
		ETC		

- Number of words in block
- Nash link. Points to next entry with same hash code.
- Rotivity bit. O if active, 1 if inactive block.
- Set if fatal error.
- Set if nonfatal error.

CNAA 1- First character in variable field.

CMAR MC-Last character in variable field.

- BDL Block data link
- CAL Common array length
- Terminating bit. Set if last set of headers in entry.
- NH Number of headers.
- SAN Starting address of headers.
- HOW Neader (pointer)

6.00.00 9- 18

Relocatable Object Code

#### Entry Type 8

PROCEOURE - SECONDARY ENTRY POINT

0 1 2 3 4		10 11	15
	HH 		8
	HL		
R   C //  H	NC	CNAA. 1	
(VARIR	BLE #CHRR.	SEE NC)	
CNAR. NC	1////	///////////////////////////////////////	///////
L I	SECF		
	SSA		
P NP		СН	
	TN		
	PARM. 1		- <b></b>
   	:		
	PRRM. NP		

- NW NUMBER OF WORDS IN ENTRY BLOCK
- HL HASH LINK POINTS TO NEXT ENTRY WITH SAME NASH CODE
- A ACTIVITY BIT. O IF ACTIVE, 1 IF INRCTIVE ENTRY
- C CRLLABILITY BIT SET IF ENTRY POINT IS UNCALLABLE
- N HIDDEN ENTRY POINT. SET IF ENTRY POINT HILL NOT BE IN LIBRRRY DIRECTORY
- NC NUMBER OF CHRARCTERS IN NAME. MAX IS 16

Relocatable Object Code

Entry Type 8 (cont.)

CHAR 1 - FIRST CHRRRCTER IN VARIRBLE LIST

CHRR NC - LAST CHRRRCTER IN VARIRBLE

L - LAST ENTRY IN LIST O IF NOT LAST 1 IF LAST

SECL - SECONDARY ENTRY POINT LIST LINK

SSR - UNIT STRATING PB' RODRESS

- P PARM CHECKER
  OO NO CHECKING (IMPLIES NP UNDEFINED,
  IN FIND PARMS ROSENT)
  O1 CHECK PROCEDURE TYPE (IMPLIES NP
  IS UNDEFINED RND PARMS ROSENT)
  10 CHECK PROCEDURE TYPE RND NUMBER
  OF PARMS (IMPLIES PARMS ROSENT)
  11 CHECK PROCEDURE TYPE NUMBER OF
  PRRMS RND TYPE OF PARM.
- NP NUMBER OF PRRMS
- CM CHARACTER COUNT OF PARMS
- TN PROCEDURE TYPE

#### Entry Header Format

SRH>	HEROER
	i . i
	HERDER
\$RH>	HEADEA
	1
	i : i
SAC>	
SHL>	CODE
	i . i
	!
	NERDER

·;	HERDER
	i . i
	HERDER

EACH ENTRY (EXCEPT SECONDARY ENTRY POINT ENTRIES) MAY DESCRIBE N> 0 SETS OF HERDERS. THE HERDERS IN EACH SET MUST BE CONTINUOUS AND IN THE SAME ORDER AS THE HON LIST DESCRIBING THE SET.

THE CODE MODULE MAY BE PLACED IN ANY POSITION IN A HEADER SET. NOTE THAT IF THE CODE MODULE IS AT THE BEGINNING OF A SET, SAC = SAN.

IF THE ENTRY HAS NO NEADER SET, THEN NN, SAN SEQUENCE IS ABSENT.

6.00.00 9- 21

#### Relocatable Object Code

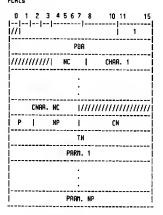
#### Neader Type 0

#### GARBAGE

0	1 10	11	15
///	HU	0	
	GARBAGE	<b></b> -	

#### Header Type 1

#### **PCRLs**



PBA - PB' ADDRESS OF LINKED LIST OF PCAL
INSTRUCTIONS TO BE REPAIRED - LOWER
14 BITS USED RS NEGATIVE DISP. - BIT O
SET MERNS THAT URDOI IS NOT A PCAL
INSTRUCTION BUT A POINTER TO A SST
LABEL OF "YEXTERNAL" FORMAT - A
LINK OF O TERMINATES THE LIST - BIT 1
SET MEANS THAT THE WORD IS TO BE

G.00.00 9- 22

#### Relocatable Object Code

INITIALIZED WITH THE PB ADDRESS OF THE PROCEDURE.

#### Header Type 2

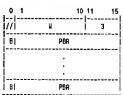
#### PR RODRESSES

0 1		10	11	15
7/	HH			2
	PBA			
	•			
1	:			
	Paa			

PBA - PB' ADDRESS OF PB ADDRESS TO BE CORRECTED

#### Header Type 3

#### OUN/ORTH VARIABLES



PBR - PB' ADDRESS OF OWN VARIABLE POINTER TO BE CORRECTED

# Relocatable Object Code

# Neader Type 4

#### DSDB/DWN/DRTR/VALUES

0 1		10	11	1	5
///	HLI		i	4	į
	LD				
B	I	H			-!
	INITIAL	VAU	UES		-      

LD - LOGICAL MORD DISPLACEMENT
IN OWN RARRY FOR INITIAL VALUES
B - BYTE BIT-SET IMPLIES THAT LD IS
TYPE BYTE AND THAT THE FIRST
WORD OF THE INITIAL VALUE BLOCK
IS A COUNT OF THE WINDER OF BYTES
IN THE INITIAL VALUE BLOCK
IN - INTERRETION NUMBER - NUMBER OF
TIMES THE BLOCK OF INITIAL VALUE
IS TO APPERA IN THE SECONDARY BD 1-MO DUPLICATION,
2->OUPLICATION, ETC

#### Neader Type 5

#### PUST

. 0	1		10	11	15
//i		NU			5
PBA					
		INITIAL VAL	.UES		

PBR - PB' RODRESS OF LINKED LIST OF POINTERS TO BE INITIALIZED WITH OB RODRESS OF PUST (SRME LIST FORMAT AS FOR FORMAT STRINGS) R PBR of -1 [WHICATES WE FIX-UPS.

NOTE: ALL REFERENCES TO THE PUST INCLUDE THE FOUR-HORD HEADER THAT IS APPENDED BY THE SEGMENTER. THESE MORDS ARE NOT PRESENT IN THE KEADER; THEY ARE AUTOMITICALLY RELOCATED AND INITIALIZED BY THE SEGMENTER.

#### Header Type 6

#### GLOBAL VARIABLES

0 1	7 8 10 11 15
// NW	6
	TN
OBA	/////// NC
CNAR.1	CHAR. 2
	:
CHAR. NC	111111111111111111111111111111111111111

#### Neader Type 7

#### EXTERNAL VARIABLES

0123456		10	11	1
- - - - -  //ן אא				7
	TN			
M ////  NC	I	CHAR.	. 1	
	•			
 	:			
CHAR. NC	1//	111111.	////	///
	OA			
	PBA			
	:			
	PBA			

PBR-PB' address of linked lists of instructions to be repair-ed; lower 3 bits of inst. used as neg. displacement to next instruction; a link of O terminates the list.

-Monitored variable bit;set if variable is being monitored by debug.

DA -Logical word disp. in PUST; lower 8 bits of word will be init. with prim.DB address of variable;DA is present if M=1.

NOTE: PBA of -1 implies null list

G.00.00 9- 25

#### Relocatable Object Code

#### Header Type 8

#### PRIMARY DB

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
U   U   U   U   U   U   U   U   U   U
:
U  U  U  U  U  ////////////////////////
INITIAL VALUES

U - AOORESS BITS
OO IF NO AODRESS
O1 IF NO RODRESS
10 IF WORD RODRESS IN SECONDARY DB
11 IF BYTE AODRESS IN SECONDARY DB

NUPOB

NOTE: INITIAL ADDRESSES THAT ARE SECONDARY DB ADDRESSES ARE O

RELATIVE (I.E., THEY ARE LOGICAL DISPLACEMENTS IN SECONDARY DB).

G.00.00 9- 26

# Relocatable Object Code

#### Neader Type 9

#### COMMON VARIABLES

- - - - -	78  -	10	11 	15   9	
	NHC				
/////// NC	ī	CHAR.	. 1		
	:				
CHAR. NC	177.	//////	///	////	
BI MI	НL				
	LO				
	DA				
	PBR				
					i i
					Ï
	PBA				
	٠				i i
8  N	ЖL				 
	LD				į
	DA				•
	PBA				
	:				
 	PBA	- <b>--</b>			1

#### Relocatable Object Code

# Neader Type 9 (cont.)

NUC - NUMBER OF WORDS IN COMMON ARRAY

NC - HUMBER OF CHARACTERS IN COMMON NAME- IF BLANK COMMON 4 COM'

- LOGICAL WORD DISP. IN PUST - LOWER 8 BITS OF WORD WILL BE INIT. WITH PRIM. DB RODRESS OF VARIABLE - NOTE DA IS PRESENT IF M = 1

- BYTE BIT O IF THE PRIMARY DB POINTER TO BE ALLOCATED AND INITIALIZED AND LD REE OF TYPE WORD 1 IF TYPE BYTE

- MONITORED VARIABLE BIT - SET IF VARIABLE IS BEING MONITORED BY DEBUG

NL - NUMBER OF ADDRESS LISTS FOR VARIABLE

LD - LOGICAL DISPLACEMENT OF VARIABLE IN COMMON ARRAY

PBR - PB' ADDRESS OF LINKED LISTS OF INSTRUCTIONS TO BE REPAIRED LOWER 8 BITS USED AS NEGATIVE DISPLACEMENT TO NEXT INSTRUCTION R LINK OF 0 TERMINATES THE LIST

PBR = -1 INDICATES NO FIX-UPS

# Neader Type 10

LOGICAL UNITS

0		10 11 15
//	8	10
	BIT MAP	į
ļ	DII NHY	į
i		

BIT MRP - BIT MAP OF LOGICAL UNITS AEFERENCEO; BIT 0 CORRESPONOS TO LU 0, ETC. (1 LESS THAN OR EQUAL TO LU LESS THAN OR EQUAL TO 99)

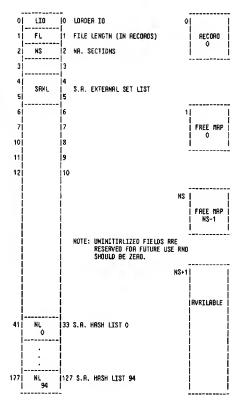
Header Type 11

FORMAT STAING

0 		10	11	15
//i N	¥		1	1
	PBA			<u>¦</u>
	MC			
CNAA. 1	ı	CHAR.	2	
[	:			-
CNRR. NC	1//	//////	////	///

PBR - PB' ADDRESS OF LINKED LIST OF POINTERS TO BE INITIRLIZED LOUER 14 BITS OF WORD USED AS NEGRITY CONSULATION OF DITTER - BIT O SET MERNS THAT THE POINTER IS TO BE TYPE BYTE - A LINK OF O TERMINATES THE LIST.

6.00.00 9- 29 RL file Format



G.00.00 9- 30

#### Relocatable Object Code

#### Storage Management

FILE SPACE IS MANAGED IN TERMS OF 32 HORDS BLOCKS (4 BLOCKS PER 128 HORD RECORD).

FREE SPACE (BLOCKS) IS ACCOUNTED FOR IN A BIT MRP, WHICH IS PARTITIONED INTO RECORDS (2K BLOCKS PER SECTION). A O INDICATES THAT A BLOCK IS USED, A 1 INDICATES THAT IT IS FREE.

FILE SPACE IS ALSO PARTITIONED INTO 512 RECORD SECTIONS (64 MAX. SECTIONS, 2N BLOCKS PER SECTION, 1 MAP PER SECTION). THE NUMBER OF SECTIONS IN A FILE IS NS=(FL+511) & LSA(9). THE FIRST NS RECORDS FOLLOWING RECORD O (RECORDS 1 TO NS) MAPE RESERVED FOR THE SECTION MAPS.

A COMPLETE FILE ROORESS WOULD HAVE THE FOLLOWING CONFIGURATION:

012345		16 26	27 31
	SECTION		OISPLEMT

FILE (WORD) ROORESS OOUBLE WORD

#### Relocatable Object Code

# Entry Point Directory

				_		
NL  >	LINK	>>	LINK	<b>&gt;</b> ,>	0	•
	USEO		USED		USEO	
				İ		i
			ļ !			İ
	//////// ////////		//////// ////////		//////////////////////////////////////	

THE DIRECTORY IS PARTITIONED INTO 95 HRSH LISTS (SAME HASH FUNCTION RS USL); EACH HASH LIST IS A LINNED LIST OF RECORDS.

ERCH RECORD CONTRINS A SUCCESSOR LINK (RECORD #) RNO R USEO SPACE COUNT. A LINK OF O TERMINATES R LIST. WHEN A RECORD IS VOID OF ENTRIES (USEO=2), ITS SPRCE IS RETURNED TO THE FREE STORAGE AREA.

# Typical Directory Entry

0 1 2 3	4567	8	15								
   S   U   I  ///	NC										
		•	į								
CHRR. NC		1//////////////////////////////////////	,,,,,,								
S.A. INFO BLOCK											
	   S.A. ENTRY										
FINI	ни со	DE									
LC   NP	ī	CN									
	TN	!									
PART. 1.											
	:		ļ								
	PAAN.	NP									

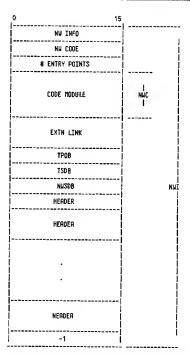
- S SECONDARY ENTRY POINT BIT SET IF THE ENTRY POINT WAS GALGINALLY A SECONDARY ENTRY POINT.
- U UNCALLABLE BIT SET IF ENTRY POINT IS UNCALLABLE.
- I PRIVILEGED MODE BIT SET IF CODE MODULE IS TO BE RUN IH PRIVILEGE MODE.
- LC is (0:2)...Level of Checking

  O = No checking

  1 >= Check for procedure type
  2 >= Check for # parameters
  3 >= Check for parameter type
  NP is (2:6) is # parameters

G.00.00 9- 33

#### Procedure Information Block



ALL HEADERS FOR THE PROCEDURE ARE APPENDED TO THE INFO BLOCK. THE HEADER SETS (EXTERHAL LISTS) ARE LINKED BY INCREASING FILE ADDRESS; A LINK OF X1777777777D TERMINATES THE LIST.

G.00.00 9- 34

Relocatable Object Code

#### Neaders

0 1 2 3 4567 8 10 11 15										
F   W   HW CODE	j									
S.A. INFO BLOCK										
S.A. ENTRY	I									
PBA										
S   U   I  ///  HC   CNRR. 1										
:	ı									
CHAR. NC  ///////////////////////////////////	<i>iii</i>									
P   NP   CN	į									
TN										
PARM. 1	PARM. 1									
·										
PARM. NP										

- F SET IF FATAL ERROR
  W SET IF HOM-FATAL ERROR
  S SATISFIED BIT SET IF EXTERHAL IS
  SATISFIED WITNIN RL.
  U WUCALLABLE BIT
  I PATVILEGED BIT

ALL HEADERS ARE THE SAME AS IN A USL EXCEPT FOR THE PCAL HEADER.

#### Prepared Object Code

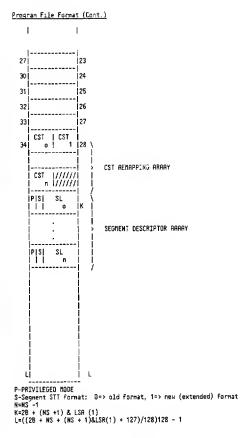
#### CHAPTER 10 PREPARED OBJECT CODE

#### Program File Format

_			
0	FLPGS	0	
- 1	NS	1	NUMBER OF CODE SEGMENTS
2	G\$	2	GLOBAL SIZE (DB TD QI) IN HORDS
3	SAG	3	GLOBAL AREA AECDAD #
4	SAS		SEGMENT SET AECDRD # (EACH SEG. STRATS IN NEW RECORD)
5	132	5	INITIAL STACK SIZE IN WORDS
6	IOF2	6	INITIAL DL SIZE IN HORDS
7	MAXD	7	MAX. DPTA SEGMENT SIZE (DL TO Z) IN WORDS
10	SAE	8	ENTRY POINT LIST RECORD #
11	SSEG	9	STARTING SEGMENT #
12	SADA	10	PRIN. ENTRY PT PB AODAESS
13	SASTLT	11	DB ADA. OF STLT (-1 IF NO STLT)
14	SAFLUT	12	(STLT=Segment Length Table) DB HOR. OF FLUT (-1 IF NO FLUT)
15	SAX	13	EXTERNAL LIST AECDAD #
16	SSTT	14	PAIN. ENTRY PT SST #
17	SATC	15	STARTING ADDRESS OF TRAPCOM'
20	SAPMAP	16	STARTING AECORD OF PMPP INFO
21	SPSI	17	STARTING AECORD OF SYMBOLIC ITEMS
22	FLAGS2	19	
23	CKSUM	19	TOTAL CHECKSUM OF ALL SEGMENTS
24		20	NOTE : ALL UNUSED WORD ARE RESERVED FOR
25		  21	FUTURE USE AND SHOULD BE SET TO ZERO.
25		22	

G.00.00 10- 1

#### Prepared Object Code



Prepared Object Code

#### Flags

9 1												
-    F	-  Z  F	- -	 //	 //	 BA :	IA	  PM	 	MR	///	DS	PH

- F FATAL EARDA IN PROGRPM

  W NON-FATAL ERROR IN PROGRAM

  Z ZEAD UNII OL RREP

  P SET IF AMY SEG IS PRIVILEGED MODE (IF NOT SET NORMAL=
  NOWPRIV MODE)

CPPABILITIES

BPTCH ACCESS (9) [BA] INTERACTIVE ACCESS (8) [IA] PRIVILEGED MODE (7) [PM]

PCCESS TO GENEARL RESDURCES

MULTIPLE RINS (4) [MR]

EXTRA DATA SEGMENT (2) [DS]

PROCESS HANDLING (1) [PM]

Prepared Object Code

#### Flags2

T - PPTCH AREA EXISTED IN ALL CODE SEGMENTS K - CHECKSUM VALIO

#### CST Remapping Array

CONTRINS THE LAST CST NUMBERS ASSIGNED TO THE SECHENTS; INDEXED BY SEGMENT NUMBER. WHEN A PROGRAM FILE IS PREPARED, THE RRANY IS INITIALIZED TO 0, 1...,N. THIS RRANY IS USED TO RE-ESTPBLISH INTRA-PROGRAM LINKAGE WHEN THE PROGRAM IS LOADED.

# Segment Descriptor Prray

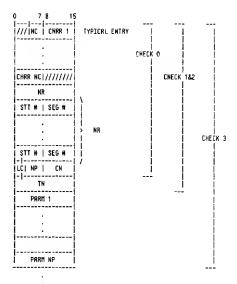
CONTPINS THE SEGMENT LENGTH AND A FLAG INDICATING IF THE SEGMENT IS TO BE LOADED IN PRIV. MODE. INDEXED BY SEGMENT NUMBER, BLL SEGMENTS BEGIN ON A RECORD BOUNDARY. THE NUMBER OF RECORDS FOR A GIVEN SEGMENT IS (SL + 127) & LSR(7). THE RECORD NUMBER, SAS, OF SEGMENT N IS

SRS:=0 FOR I=0 TO N-1 BEGIN SPS:=SRS + (SL(I) + 127)&LSA(7) EN0

#### Global Area Format

P SET OF AECOROS CONTAINING THE INITIAL VALUES FOR THE GLOBAL AREA OF THE DATA SEGMENT. THIS SET BEGINS AT RECORDS SAG (WORD 3) AND CONSISTS OF (GS + 127) & LSA(7) RECORDS.

#### External list



LIST TERMINATER

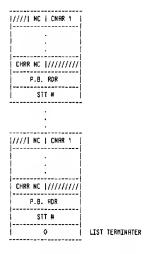
- LC (0:2) = LEVEL OF CHECKING
  0 = NO CHECKING
  1 >= CNECK FOR PROCEDURE TYPE
  2 >= CNECK FOR # PRRAILTERS
  3 >= CNECK FOR PRRAILER TYPE

= NUMBER OF REFERENCES NP (2:6) = NUMBER OF PRRAMETERS

G.00.00 10- 5

Prepared Object Code

#### Entry Point List



NOTE THAT THE ENTRY POINT LIST MUST IMMEDIATELY FOLLOW THE EXTERNAL LIST.

G.00.00 10- 6

Prepared Object Code

### Code Segment With Patch Area



# Patch Rrea

		_	
1	PROGRAM Name	4-WORD	PROGRAM NAME
	SEGMENT NAME	8-#0RD	SEGMENT NRME
	//	1-₩ORD	UNUSED
-	CHECKSUM	1-#ORD	CHECKSUM
-	PREP TIME	2-WORD	PREP TIME
įF	ATCH TIME	2-#DRD	PRTCH TIME
1	PRTCH RRER		
-	PRLEN	1-WORD	PATCH AREA LENGTH
	STT	  -  -	

Prepared Object Code

### PMRP Information

PTT	PMOP TYPE TRBLE
SPP	SEGMENT PHAP POINTER
APD	ACTUAL PMAP DRTUM

#### PMRP Type Table

PTTL	TYPE TABLE LENGTH
LPR0	LENGTH OF PHAP RECORD TYPE O
LPR1	LENGTH OF PMAP RECORD TYPE 1
:	
LPRn	LENGTH OF PMAP RECORD TYPE n

NOTE : = PTTL - 2

# Prepared Object Code

#### PMRP Records

Type O Segment PMAP Record

0 1 2 3	4 5 6 7	8901	2 3 4 5
	O! NC	l ch	ar 1
1			į
char	NC	1///////	<i>,,,,,,,</i>
STT	LEN	I SEG	NUM
	SEG LEN	STM	

Type 1 Procedure PMRP Record

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
1  NC   char 1
:
char MC \//////////
N(////////////////////////////////////
SA OF CODE
COOE LENGTH
PRIMARY ENTRY POINT ROOR
COBOL TOOL BOX ID
TOOL BOX PROCEDURE ID

Prepared Object Code

Type 2 Secondary Entry PMAP Record

• •
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
2  NC   char 1
:
char NC  ///////////////////////////////////
H1////////////////////////////////////
SECONORRY ENTRY POINT ROOR
NUMBER OF ENTRY POINTS

N : NIDDEN ENTRY FLAG

G.00.00 10- 9 G.00.00 10- 10

' Prepared Object Code

		SL File Format
0	∐0 j	
		1 FILE LENGTH (IN RECORDS)
2		2 EXTENT LENGTH (IN RECORDS)
3		3
4		4 # SEGMENTS
5		5
6		6
7	FRTL	7 S.R. DF FREE A.T. ENTRY LIST (-1 IF HONE)
10		8
11		9 # REFERENCE TABLE ENTRIES
12		j10
13		11 # SECTIONS
14		12
	i I	
	i I	] 
	 	] 
	1	
		<u> </u>
	ĺ	i I
	Ì	i I
	į	
41	   HLO	
71		1
	:	   NOTE:
4.77		SHROEO AND UNINITIALIZED FIELDS ARE
	' HL94 	

Prepared Object Code

St File Format (Cont.)

0-     	RECORO I				
1-       	RECORO I	< Ri	EFERENCE	TABLE	POINTERS
2-	FAEE MRP				
HS+1	FREE MRP     HS-1				

S+2-----

#### Storage Management

FILE SRRCE IS MRNAGED IN TEAMS OF 128 HORD BLOCKS (1 BLOCK PEA 128 HORD

FREE SPRCE (BLOCKS) IS ACCOUNTED FOR IN A BIT MAP, WHICH IS PRATITIONED INTO RECORDS (2% BLOCKS PER SECTION). A O INDICATES THAT A BLOCK IS USED; R 1 INDICATES THAT IT IS FREE.

FILE SPACE IS ALSO PARTITIONED INTO 2048 RECORD SECTIONS (16 MAX. SECTIONS, 2K BLDCKS PER SECTION 1 MAP RER SECTION). THE NUMBER OF SECTIONS IN A FILE IS NS-(FL + 2047) & LSA(7). THE FIRST MS RECORDS FOLLOWING RECORDS 0, 1 (RECDADS 2 TO MS+1) RRE RESERVED FOR THE SECTION MRPS.

IF THE SECTION MAPS SPECIFY MORE SPACE THRM IS POTENTIALLY AVAILABLE, THOSE RECORDS BEYOND FLIMIT ARE MARKED AS "USED".

#### Entry Point Directory

HL  >	LINK	->>	LINX	->>	0	ĺ
	<del>-</del>					
	USED		USED		USED	ı
						ı
	///////////////////////////////////////		///////////////////////////////////////		/////////	
	/////////		11/11/11	į į	////////	

THE DIRECTORY IS PARTITIONED INTO 95 HRSH LISTS (SRME HRSH FUNCTION AS USL); ERCH HASH LIST IS A LINXED LIST OF AECORDS.

ERCH RECORD CONTRINS R SUCCESSOR LINX (RECORD #) AND R USED SRACE COUNT. R LINK OF O TERMINRTES R LIST. WHEN A RECORD IS VOID OF ENTRIES (USED=2), IIS SRRCE IS RETURNED TO THE FREE STORAGE AREA.

THE HASH LIST HEAD POINTERS (HL IN THE DIAGRAM RBOVE) RAE IN AECORO O WOADS X41 TO X177.

Rregared Object Code

# Code Segment Linkage Structure



ERCH CODE SEGMENT OCCURIES RN INTEGRAL NUMBER OF RECORDS. THIS BLOCK OF INFORMATION CAN BE SUBDIVIDED INTO THREE TABLES: THE CODE SEGMENT RRORER, RN STT SEGMENT MRR RRARY, RNO RN EXTERNAL LIST.

R 1 BYTE X 256 BYTE RRRRY. IT IS INDEXED BY SIT HUNBER RND RETURNS (IF THE SIT CORRESPONDS TO RN EXTERNAL OF THE SEGMENT) THE SEGMENT NUMBER OF THE EXTERNAL RND 255 DIMERUISE. THIS RAMBAY IS USED WHENEVER THE SEGMENT IS LDROED AND IS URCRITED WHENEVER THE SLIS BOUND BY THE SEGMENTER.

#### EXTERNAL LIST

A SYMBOLIC LIST OF THE EXTERNALS OF THE SEGMENT. ERCH ENTRY CONTRINS INFORMATION RBOUT THE EXTERNAL: RRABNETER CHECKING LEVEL RND PARRMETER MITCHING INFORMATION, AND THE SEGMENT NUMBER AND STI NUMBER IF THE EXTERNAL IS SATISFIED WITHIN THE SL.

Prepared Object Code

#### Typical Directory Entry

٥	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
///\	U	1///1	R I		HC	:	ا				CHAR	1			
				- <b></b> -		. <b>-</b>	:								
	<u>.</u>	CHRR	NC					///	////	////	////	////	////	////	////
		\$11	# 		<b></b>		1			S	EG W				
LE		l 		NR			I				CH				
<b>-</b>							TN								
							RARM	1					<b>-</b>		
							<u>:</u>								- <b>-</b>
						<b></b>	PRRM	NP	<b>-</b>						

- LC is (0:2)...Level of Checking
  0 = No checking
  1 >= Check for procedure type
  2 >= Check for # parameters
  3 >= Check for parameter type
  NR is (2:6) is # garameters
- R 0= Not permanently allocated 1= Permanently allocated
- U Uncallable bit set if entry point is uncallable.

G.00.00 10- 14

Rrepared Object Code

0 1 2 3 4567 8

Code Segment Structure (Cont.)

<u>|-|-|-|-|----|</u>

15

CODE SEGNENT	
STT MRR ARRRY	ļ
  S / / /  NC   CHRR. 1	s -
CHRR. NC  /////////	
STT #   SEG. #	
P I NP I CN	ļ
TN	
PRRM. 1	į
	İ
PARM. NP	İ
	ļ

SATISFIED BIT - SET IF EXTERNAL IS SRTISFIED WITHIN SL

EXTERNAL LIST TERMINATOR

#### Prepared Object Code

#### Reference Table Structure

FOR ERCH SEGMENT THERE IS R REFERENCE TRBLE ENTRY OF 32 MORDS. THE REFERENCE TRBLE ENTRIES RRE PRCKED FOUR TO R RECORD. THE RECORDS CONTRINING THE REFERENCE TRBLE ENTRIES ARE LISTED IN RECORD 1. THE RECORD CONTRINING REFERENCE TRBLE ENTRY IS REC 1 (N.(0:14)); THE FIRST MORD OF THE ENTRY IS REFTRB (N.(14:2) & LSL (5)).

WHEN R SEGMENT IS DELETED, THE REFERENCE TRBLE ENTRY CORRESPONDING TO THE SEGMENT IS RELERSED. THESE FREE ENTRIES ARE LINKED TOGETNER IN R LIST; THE SEGMENT # IS USED RS R LINK RND IS PLRCED IN THE FIRST WORD OF THE ENTRY.

UNEN'R SEGMENT IS RODED IT IS ASSIGNED R SEGMENT NUMBER (O LESS THRN/EQURL TO N LESS THRN/EQURL TO 254); THE NUMBER IS THRT OF THE FIRST FREE REFERENCE TRBLE ENTRY, OR, IF NOME RRE FREE, THE NEXT AVRILABLE REFERENCE TABLE ENTRY (CRUSING SPRCE RELOCRTION FOR THE ENTRY).

#### Prepared Object Code

# Reference Table (256 Maximum Entries)

#### TYPICAL ENTRY

OREC. 1		R.T. REC.		0 1 2 3 4 5 6 7 8 9 15	X
RL	>	E 0	 	PINI SEGMENT LENGTH	0
		E		SEGMENT RODRESS (REC. #)	1
1:		1	1	# REC'S FOR SEG. & EXTM. LIST	2
:		E 2	į	F S / / R C X / /  # ENTRY PTS.	3
		<u>-</u>	į	SRPMRP	4
63		3	l	SRSI	5
(FILE RECT	)	(1 SECTOR)		TIKI	6
		THE RRRRY		SI LENGTN	7
1	ER I	ND CNRRRC- CDUNT RNO LING BLANKS O.		SEGMENT NAME	10
Đ: Ri	INDE: IT SI FERI	BIT ARRAY KED BY SEGH ET IF SEG I ENCEO DIREC INDIRECTLY	Ś T-	SCOTENT MINE	20
R PERMRI C CORE I X MPE SI	NRL NENT RESI EGNE INST FLAG FLAG	SRTISFIED LY RLLOCRTE DENT SEGMEN NI . IN SEGMEN G	T	REFERENCEO SEGMENTS BIT HRP	
-	=>	SEG STT IS OLD FORMRT SEG STT IS NEW FORMRT EXTENDED CS	IH		

G.00.00 10- 18

G.00.00 10- 17

Prepared Object Code

Code Segment With Patch Rrea

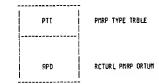
PRTCH RRER

Patch Rrea (Cont.)

•	SEGMENT     NRME	8-MORO SEGMENT NRME
	//	1-WORD UNUSEO
	CHECKSUM	1-WORD CHECKSUM
	PREP TIME	2-WORD PREP TIME
	PRTCN TIME	2-MORO PRICH TIME
	PRTCH RREA	
	PRLEN	1-MORO PRICH RRER LENGTH
	   \$17	

Prepared Object Code

PMRP Information



PMRP Type Table

	PTTL	TYPE TRBLE LENGTH
į	LPRO [	LENGTH OF PMRP RECORD TYPE O
	LPR1	LENGTH OF PMRP RECORD TYPE 1
	:	
	LPRn	LENGTH OF PHAP RECORD TYPE n

NOTE: n = PTTL - 2

#### PMAP Records

Type 0 Segment PMRP Record

0 1 2	3 4 5	678	9 0	1 2	3	4	5
!	01	HC	1	char	1		
							-
		:					
char	NC	1/	////	////	//	///	11
STT	LEH	ı	SE	G HU	n		-
	SEG	LENGTH					

Type 1 Procedure PMAP Record

#### 0123456789012345

0 ( 2 3 4 5 6 / 6 5 0 1 2 3 4 5
1  NC   char1
:
char NC  ///////////
H1////////////////////////////////////
SA OF CODE
COOE LENGTH
PRIMARY ENTRY POINT ACOR
COBOL TOOL BOX 10
TOOL BOX PROCEOURE IO

Type 2 Secondary Entry PMAP Record

0123456789012345

	2  H	IC	c	har 1	
		:			
char I	i¢	1//	////	/////	///
H[/////	/////	////	1111	/////	///
SECONO	ARY E	NTRY	POIN	T AOOF	
IUN	BER C	F ENT	RY P	OINTS	

H : HIDDEN ENTRY FLAG

G.00,00 10- 21

G.00.00 10- 22

#### CHAPTER 11 LORDER

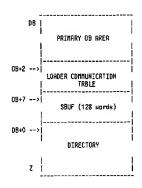
#### MPE Loader

The loader is a system process which will do loads sequentially. If a process needs code to be loaded, it will get the load process' SIR, fill loader communication table, and then awake the loader. Upon completion, the loader will return its status through the loader communication table, and then activate the waiting process.

#### Loader Segment Table Overview

Loader Segment Table consists of two OST's. The main one is DST X22 (LST). The other DST (XLST) has its DST number stored in SYSGLOB X226.

#### LS! Overview



Loader

#### KLST Dveriex



The above DST's has exactly the same primary DB area so that directory entry handling procedures can be used on both DST'S. XLST is the LST extension and is used to store the extension entry only. When a extension entry is needed, it is copied into the LST to elininate frequent EXCHRNGEOB. Note that XLST is capable for any types of entries. It is used for extension entry only for now. Also, some of the primary OB's in the XLST are not used. They are there just for the consistency.

G.00.00 11- 1

G.00.00 11- 2

#### Loader

#### Loader Segment Table Primary D8

0	@DIR	16	SD
1	DIR LEN	17	SP
2	@LCT	20	SQ
3	ENTP	21	SR
4	ENTP1	22	88
5	ENTP2	23	ST
6	ENTP3	24	HDFWLINK(TYPE 0)
7	@SBUF		:
10	21		HDFULINK(TYPE 8)
11	SJ		NOBKLINK(TYPE O)
12	SK	\	 
13	SL		HOBKLINK(TYPE 8)
14	Sm		LCT
15	SN		:
		i	

ENTPM : POINTERS POINT TO THE CURRENT RCCESSED ENTRY.
SBUF : UTILITY BUFFER. USUALLY CONTRINS PROGRAM FILE RECORD
O INFORMATION.
SI ST: UTILITY OB RELATIVE VARIABLES.
HOFULINKS : MERD OF FORWARD LINK FOR ERCH TYPE.
HOBKLINKS : HERD OF BRCKWARD LINK FOR ERCH TYPE.

#### Loader

# Directory Entries

0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  	GRRBRGE(O)
BACKWARD LINK	
LENGTH	
0	
GARBRGE	
O  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  	SL FILE(1)
BACKWARD LINK	
LENGTH	
1 1	
FILE DISC RODRESS	
FILE PV INFD	
# RLLOCRTED SEG   # SEGLIST ENTRIES	
SEG RRRRY ( 16 WORDS )	
LDG SEG NUMBER    A  C  X  M	   SEGLIST RRRRY
REFERENCE COUNT	> 3 NDRO ENTRY    PER ALLOCATED
PHYSICRL CST NUMBER	I SL SEG
:	′
!	
[	

#### Directory Entries (Cont.)

0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 	PRDGRAM FILE (2)
BACKWAAD LINK	
LENGTH	
PIA I LIB   2	
FILE DISC ACCAESS	
CST BLOCK INDEX	
SEGNAP DST	
# PRDCESS SHRRING	
N SEG IN PAOGRAM FILE   # SLINFD AREA	
PV FILE INFO	
TRACE EXTEANAL PLABEL	
SL SERRCH SEQUENCE	
SL FILE DISC RDDRESS	SL INFO ARER > 19 NORD PER   EACH SL FILE
LIB SEG AARRY (16 WORDS)	,
:	
:	
PSEGMAP SIZE	)
LIB LOG SEG   SL INFO IMDEX	
LIB LOG SEG   SL INFD INDEX	PSEGNAP > AARAY
; ;	
LIB LDG SEG   SL INFD INDEX	}
G.00.00 11- 5	

Directory Entries (Cont.)

0  1  2  3  4  5  6         Forwerd	71 81 91     LINK	10 11 12  	13 14 15  	LORDING(3)
BRCKURRD	LINK			
LENGTH	<b></b>			
P	ı	3		
FILE DISC	RDDRESS			

-	4  5  6  7  8  9  -      FDAWAAD LINK	10 11 12	13 14 15	WRITER(4)
	BRCKWRRD LINK			
	LENGTH	··		
PI		4		
FI	LE DISC ADDAESS			
	URITING PIN	·		
	UNUSED			
,				

G.00.00 11- 6

Directory Entries (Cont.)

Loader

LORDED(5)

Loader

# Directory Entries (Cont.)

	LENGTH		į	
Ρļ	1	5		
ı	TLE DISC MODRESS			
	LOAD PADCESS ST	RTUS	   	
D  1  2  3  	4  5  6  7  8  9        FDANAAO LINK	10   11   12	13 14 15  	SHAREA(6
	BRCKWAAO LINK		···	
	LENGTH			
PΙ	l	6		
	PIN			
F	TLE DISC RODRESS			

3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15 -|--|--|--|--|--|--|--|--|--|--| FORLMARO LINK

EXTENSION(7) LENGTH | LIB | PIN EXTENSION ID LOADPROC COUNT(LOADPROC)/LOG SEG#(RLLOCATEPAGC) PLABEL # CHAR IN NRME | PROCEOURE NAME | N SL INFO AREA SL INFO AREA (19 NDAOS PER SL INFO ENTRY) MCSTREFSIZE MCSTIDX(1) MESTREF RRRRY MCSTIOX(H)

18

19

20

21

RECEBUNT

NAME

PV INFO

G.00.00

17

18

19

20

21

RECOUNT

PV INFO

G.00 00

#### LCT (Cont.)

# form Returned (No Error)

0	n Infl Starting Segment Numbea
1	0
2	LOAG MAP FLAG
3	LDEY
4	DISC
5	ADDRESS
6	TRACE LABEL (IF TRACE)

#### form Returned (Error Occurred)

0	FILE SYSTEM ERROR #	ı
1	LOADEA EARDR #	i
		ı

6.00.00 11- 13

# Logical Segment Transform Table (LST!)

When a process references any user SL segments, these segments are assigned logical segment numbers if the new mapping ucode is running. The LSTI provides a map mapping these logical segments into their physical segment numbers and having true STIT's for the mapped segments. The LSTI is created by LOMDER during the load time. It occupies an OST and the DST number is stored in PEB(15), if no user SL segment is referenced, the LSTI will not be meeded, hence it will not be created.

The new mapping nicrocode depends on the existence of the LSTI for getting the physical segment number for a mapped segment. So the LSTI has to be included in process' locality list if there is an LSTI. Dispatcher will then bring the LSTI in before the process can be num. Also the bank and address for the LSTI belonging to the current running process are stored in sysglob cells ( X221 and X222 ) during the launch time by the dispatcher. These cells are used by microcode for fast accessing the LSTI.

G.00.00 11- 14

Loader

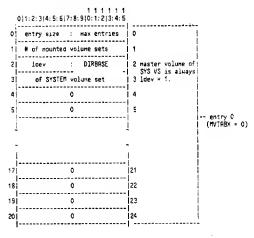
#### Logical Segment Transform Table (LSTT) (Cont.)

	į	•	of Log	ical Seg	mente	į	
	į		Lengt	h of LS	П	į	
	i		Phys	icel Se	nent #	1	Logical eeg 1
	·		Pointer	to STT	list	ĺ	Codicat sed .
ļ	i		Phy	ical Se	gment #	į	Logical meg 2
-			Pointer	to STT	list	<u> </u>	Engicer seg t
- [	i			•		į	•
-				:		i	
ļ		i	Phy	sical Se	gnent #	i	Logical seg m
	-	• 	Pointer	to STT	(Max 255)		
		H	STT #	<u>-</u>	SEG #		STT'e for logical
		H	STT #		SEG #		eegnent 1 (if needed)
		• Į					(11 useosa)
$\Pi$		 +					•
11		n			SEG #		:
	->	} +	Total	STT's	for this	eeg	i •
Ì		-		:			:
İ		ļ					l
į		İnl	STT N	<u> </u>	SEG #		+ STT'e for logical
i		ini	\$17 .	1	SEG #		segment n + (if needed)
į		İ					1
i		;   m :	* 172	<u>-</u> -	SEG #		• 
i	->	†			for this		<b>†</b>
		·				<del>-</del>	+

#### Private Volumes/Serial Disc

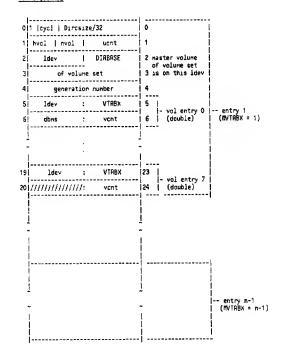
#### CHAPTER 12 PRIVATE VOLUMES / SERIAL DISC

#### Mounted Volume Table (MVT AB) DET =53 (X65)



#### Private Volumes/Serial Disc

#### MVTAB (Cont.)

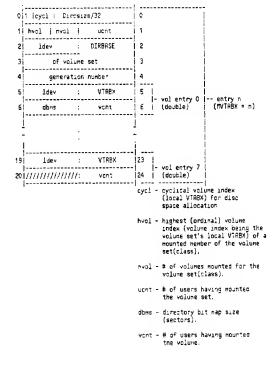


G. 00.00 12~ 1

G.00.00 12- 2

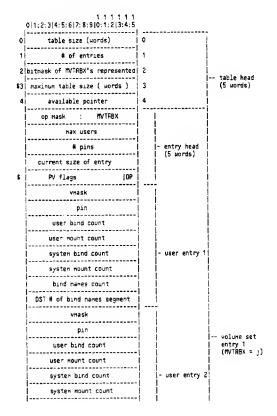
#### Private Volumes/Serial Disc

#### MVTRE (Cont.)



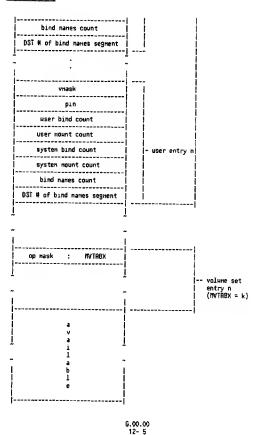
#### Private Volumes/Serial Disc

# Private Volume User Table (PVUSER) DST =54 (66 )



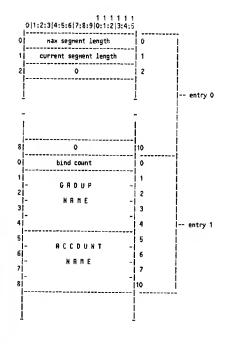
0.00 00

#### PVUSER (Cont.)



#### Bind Wanes Data Segment

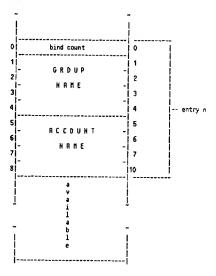
(Created and managed via PVUSER Table)



G.00.00

# Private Volumes/Serial Disc

#### Bind Hames Data Segment (Cont.)



#### Private Volumes/Serial Disc

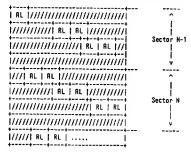
#### Serial Disc Tables and Data Structures

#### Data Record format

The primary purpose of the Serial Disc Interface (SDISC) is to adapt the undefined length transfers characteristic of magnetic tape to the fixed-length environment of a disc or integrated cartridge tape(ICT). To accomplish this, data is buffered uithin SDISC. The buffer is an integral member of sectors (blocks for the ICT) long. Files always start on a sector boundary, but data records uithin files may start anywhere and straddle sector boundaries. A record in the buffer is structured as follows:

record	record   length
(bytes)   	(bytes)

The record length is always a one-word positive byte count which includes only the data portion of the record, not the length words themselves. Records within a file might be stored on the disc as follows:



The reason for the trailing byte count is to implement an easy way to backspace records.

#### Private Volumes/Serial Disc

#### End of File Format

Since files always start on a sector boundary, it follows that they also nd on one. End of files consist of a O record length and O-fill to the end the current sector as follows:

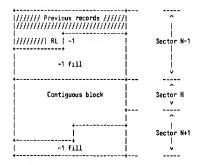


In addition, an End-of-File entry is made in the Gap Table, so that files may be skipped by scanning Gap Table entries instead of serially scanning the data area. The Gap Table is described a feu pages from now.

Private Volumes/Serial Disc

#### Contiguous Block Format

A serial disc, if it can do everything a magnetic tape can do, must also be a cold-load device. This means that machine microcode must be able to read a bootstrap channel program and the resident segments of INITER from the disc into memory. The microcode and channel programs cannot deal with the record length words which surround standard data records, so for them we have a structure, called a CONTIGUOUS BLOCK, which has the data without the length words. Information as to the length of each contiguous block must therefore be kept elsewhere, so there are Gap Table entries which hold the beginning and ending sector addresses of each contiguous block. This implies that each block must begin and end on a sector boundary. In this way they are similar to data files. To set contiguous blocks off from normal data, and to reach a sector boundary, a record length and fill character = %177777 is used, as follows:



#### Hole Format

Holes on the serial disc have the same format as contiguous blocks (that is, they start and end on sector boundaries with -1 fill characters as required). Starting uith NPE version G.00.00, holes are obsolete and SDISC will not generate them. However, code has been left in SDISC to process any holes found on serial discs written with earlier versions of SDISC. Further details may be found in the Serial Disc IMS.

G.00.00 12- 10

G.00.00 12- 9

#### Private Volumes/Serial Disc

#### Gap Table Format

The Gap Table is a four-word header followed by a series of two-word device address entries. A permanent copy lives on the device, starting in sector 4, while a working copy lives in main menory. The copy in memory is posted to the disc only when a backspace or rewind operation occurs after writing (in other words, when the copy in main memory has changed). The length of the Gap Table is device-dependent according to the table below:

Device	Humber of sectors (or ICT blocks)
HP7920	44
HP7925	106
HP7933/35	219
HP7902/9895	26
TCT	4 blocks ("S" cartridge) or
	15 blocks ("L" cartridge)

The Gap Table looks like this:



The type field is bits 0, 1 and 2 of the first word. The eight possible types are:

- O. End of File. The associated sector address contains one or more end of file fill characters (O) to fill out that sector. In the worst case (the previous record ended exactly at the end of the previous sector), the end of file sector contains all zeros.

  1. End of data. The associated sector address is the last address of valid data plus 1, in other words, the next available address. In practice, such an entry is usually preceded by an end-of-file entry, since the EDD entry is written when you stop writing, and the file system will not let you backspace or rewind after writing without sending a Write End of File. An EDD entry is also written at the beginning of the Gap Table when new (unwritten) media is inserted. This prevents erroneous reading of blank media.

#### Private Volumes/Serial Disc

- Beginning of Hole. The starting address of a "defective" area of the disc. Usually on a track boundary, but may be in mid-track if a contiguous block was being mritten when the "defect" was encountered. Dbsolete, starting with MPE version G.00.00. End of Hole. The corresponding ending address of the "defective" area. Always at a track boundary. Obsolete, starting with MPE version G.00.00. Beginning of (contiguous) Block. The starting address of a contiguous block, exclusive of the -1 fill characters which nay have been required to get us to a sector boundary. Unlike the End of File fill characters, there need not be any -1 characters if the previous record or contiguous block (with or without the trailing length word) ended exactly on a sector boundary. End of (contiguous) Block. The address of the last sector containing contiguous block data. The sector nay also contain -1 fill characters to get us to a sector boundary, but as with the beginning of block they are not required if the contiguous block ends exactly on a sector boundary. End of Tape nark. The sector address of the simulated End of Tape reflector. This type is nou written only to floppy discs for use by INITIBL's serial disc interface. When read by MPE's SDISC, it will be skipped no hatter unhat device it is found on. This ensures compatibility uith older serial discs.
  End of Gap Table. No associated sector address. If you hit this while scanning the Gap Table, you've gone too far. In practice, this type is created whenever the Gap Table is cleared, by the simple device of initializing the table to -1.

#### Private Volumes/Serial Disc

#### SDISC Extra Data Segments

With insignificant exceptions, SDISC operates entirely in split-stack mode, that is, using an extra data segment for its working storage. Starting with MPE version G.OO.OO, there are two additional data segments used as no-wait data buffers. For the nost part, our discussion here is restricted to the original data segment, now used only for variables, the Gap Table, and data buffer namagement.

original data segment, now used only for variables, the Gap Table, and data buffer management.

The working storage extra data segment (XDS) is usually acquired by the external procedure BLLDCRTE when the serial disc device is first assigned to a user as part of an FDPEN. The external procedure DERLLORIT makes the XDS go away as part of its processing of the final FCLDSE against the device. The system program PVPRDC may also acquire and release an XDS so that the tape label routines in LRBSEG nay also use SDISC for their work when DEVREC processes a device on-line interrupt. SDISC allocates the two data buffer segments as they are needed, then deallocates then as part of the Devrectiose processing.

In addition to the Gap Table already described, the XDS contains SDISC's global storage area, including the data buffer management areas (BUFFER/IMFD), and a small buffer (called UDKKTRBLE). DUKKTRBLE bolds the contents of the Serial Disc label sector when SDISC reads it in as part of its self-configuration. It also hold the Defective Tracks Table (TRC family discs) or Defective Sector Table (CS80 discs) while reassigning suspect or deleted tracks.

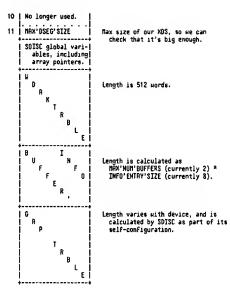
The three arrays in the XDS (UDRKTRBLE, BUFFER/INFO and GPT (Gap Table)) are all dynamically configured by SDISC as vanilla indirect arrays, such as might have been constructed by SPL. This is done by declaring the array names as pointers, then inserting appropriately computed element-O addresses in them.

#### The extra data segment is organized as follows:

0	WDADSPERSECTR	These twelve words are reserved for use by RLLOCATE when the data
1	SECTORSPEATAAK	segment is created. However, AL- LOCRTE only stuffs the last five
2	STARTADORESS (BDT)	of them. We fill the first seven ourselves with information we get
3	EDTSECTA (disc	from the label sector.
4	lated end of tape)	
5	EDDSECTA (last     sector of disc)	Simulates tape runoff.
6		oxidates tape inform.
7	JUSTALLOCATED	Tells us to initialize SDISC
8	WRITE RING	parameters to BDT if true. Simulation of tape write ring.
9	FATRLERROR	Disables SDISC when true.
	1	

G.00.00 12- 13

#### Private Volumes/Serial Disc



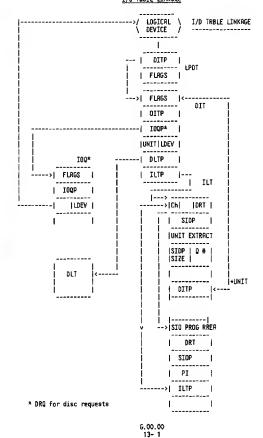
G.00.00 12- 14

#### Private Volumes/Serial Disc

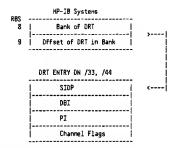
#### Serial Disc Dryanization

#### The disc is organized as follows:

A	
Label sector	O See expanded view in Chapter 3.
DTT/DSCT	1 DTT (MAC family) or DSCT (CS80).
Cold load	2 HP-IB cold load channel prog.
Soft dump	3 SDFTDUMP channel program.
Gap Table   .	4 to STARTRODRESS - 1.
†   Data	TARTADDRESS
Vala	I STRATRODALSS
i .	· '
	to
	EDTSECTA
	to
Last data sector	EODSECTA



#### Device Reference Table (DRT)



SIDP - absolute address of SID program PI - interrupt handler plabel DBI - this is the absolute address of the ILT

G.00.00 13- 2

I / 0

#### Driver Linkage Table (DLT)

٥	O 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 	DPROC
1	MONITOR PLABEL	OMNTR
2	INITIATOR PLABEL	DINIT
3	COMPLETOR PLABEL	DCDMP
4	INTERRUPT PLABEL	DINTP
5	OIT SIZE   DEVICE TYPE	DTYPE
6	CS DRIVER EDITOR PLABEL	
7	INITIRLIZATION PLABEL	

There is one DLT for each type of driver. A pointer in the OIT allows different devices on a controller to have different drivers and interrupt handlers.

DPROC. GNUMB - This field contains the I/O process request queue number for type 2 drivers. Zero for all other types.

(8:1).DRVRFRZN - Driver code frozen. Set by MRM when then the driver code segment has been made present and frozen from a request from SIDOM.

(9:1).MRMERRORC- MRM Error on Code Makepresent

(MC) .(10:1).CORERES - If set both initiator and completor code are core (.10:1).CORERES - IF set both initiator and completor code are core
(CR) resident.
(.14:2).DAVRIYPE- DRIVER/MDHITDR TYPE

0 - not used
1 - driver can be executed on any stack
2 - driver can be executed in the user process or
in the I/D process identified by IDNUMB
3 - run only in process whose PCB number is in
IDNUMB - I/O Monitor Plate

DMNTR - I/O Monitor Plabel.
DINIT - Driver Initiator Procedure Plabel.

OCDMP - Driver Completor Procedure Plabel.

DINIP - Special interrupt handler Plabel. This procedure is called by GIP if ISPEC is set DFLMG. No other action is taken by GIP except to set the Interrupt Status in DSTRI.

DTYPE.DITSIZE - The length of the DII in words for this driver.

I / 0

#### Logical-To-Physical Device Table (LPDT)

DST = 13 (= 215) SIR = 9 (= 211)

The LPDT has several fields which describe the state of a device. Some of these fields have the same meaning for all devices. Others are device dependent. Rll are described below.

There are two types of devices represented in the LPDT: real devices and virtual devices. R real device is one which has been configured into the system and is capable of performing input and/or output. A virtual device simulates some of the properties of a real device (for example a spooled line printer or an IMP), but there is no physical I/O involved. The two main uses for virtual devices are for OPEN spooled devicefiles and certain conmunication devices (such as IMP's). A given virtual device entry is in use only while the devicefile it represents is open. When the file is FCLDSCd, the entry becomes available for another virtual device. This is the reason for the SYSDUMP/INITIAL configurator question MRM % 0F OPEN PROUNTIES—it needs to know how many virtual device entries to allocate to the LPDT (and to the LOT). Entries in the LPDT are ordered by logical device number. The first word address of a real device entry is obtained by multiplying the LDN by the entry size. Except for the Oth entry, entries for which no logical device is configured on a given system are used for virtual device entries. Any remaining virtual device entries follow the last real device entry.

#### Entry 0

٥	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1	Entry size = 4
2	DEVAEC service request count
3	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

Discussion:
Word 2 is incremented by a device driver whenever it sets the Device
Ownership State field (below) to 2 (Service Requested). DEVREC decrements the
count for each interrupt it services until the count reaches 0, at which time

-- CRUTION --Device drivers must lock this table by DIS-RBLE/ENRBLEing, -NDT- by trying to acquire the LPDT SIR.

#### Typical Entry (Virtual Devices)

	٥,	1																							
0	1				•	- (	Poi	nt	e	•	to	1	ÖC	) ;	ul	be	n1	r	y			Ī			
1	 	i		Ĺ	i		ĺ	Ĺ	i	i				i		i		i							i
	///							11	11	1	11	11		11	11										
3	101	///	//	//	//	//	///									1	//	1	//	/	//	7	//	//	/

ID -- O for input, 1 for output.

Mord O, bit O is 1 for a virtual device, O for a real device. The fields in word 1 are the same, as applicable, as for the real device represented by a given virtual device. See below.

G.00.00 13- 5

I/D

I / 0

#### Entry for Terminal-Like Devices

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 +							
Device   JDa Ct  D  I  End of   B  L  Device   1 Ounce  olta IY  u  n  File   r  o  Subtype    State  b    p  t Cndition  k  g							
2  SYSDB-relative pointer to the DIY							
3.1111111111111111111111111111111111111							

Discussion (unique fields only):
Word 1.(4:1) -- CONTRDL-Y is allowed and has been detected.

Word 1.(10:1) -- BREAK has been detected -OR- ignore BREAK if the C.I. is

Word 1.(11:1) -- The terminal is logging on. This bit is set by PRDGEN and DEVREC when the logon sequence etarts. If the bit is off when polled by INIIJSMP, the terminal has disconnected. For now, only IDTERNO and MIDTERN support the use of this bit. Multipoint and DS pseudo-terminals do not.

#### Entry for Tape Drives

0 1 +   0  0 ///	 /////	  /////	11-	MIIII.	 //////		
Devc   1 Duned	J Da o ta bi	B  C   0  u   T  p	I I I	End of File ndition	A     V     R	Au: De to: Su	
2				pointer		DIT	
3/////	/////	/////	/////	//////	/////	(//////	11111

Discussion (unique fields only):

Word 1.( 4:1) -- BDT. Tape is at load Point -DR- no tape nounted. Recording density may only be suitched when this bit is true (for multiple density tape drives).

Hord 1.(11:1) -- If true, DEVREC is performing Automatic Volume Recognition (RVR) on a tape (or PVPROC is doing the same on a serial disc), -DR- RVR is to be suppressed on job or data accepting devices.

Typical Entry (Rll Real Devices)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | Devc | J|Da | D| I| End of | 1 | 1|Duned | o|ta | u| n| File | 1 | State | b| | p| t|Cndition | 1 Hu: Device Ito: Subtype SYSDB-relative pointer to the DIT 

Discussion:

Word 1.( 0:2) -- Device Ownership State:

0 -- Not owned by any process.

1 -- Duned by a process.

2 -- Service requested. Set by driver for unexpected interrupt, then wakes DEV-REC.

2 -- Service granted. Set by DEVREC. Logon unexpected interrupt, then wakes DEVREC.

3 -- Service granted. Set by DEVREC. Logon
sequence is 0-2-3-1.

3 -- Device reserved (alternate use). Set
during STARTSPOOL, spooler process
sets to 1 when it gets started.

Word 1.(2:1) -- Device is Dubinsein Riccepting if true.
Word 1.(3:1) -- Device is Data Riccepting if true.
Word 1.(5:1) -- Device is Dubincative if true (all devices except discs).
Word 1.(6:1) -- Device is Interactive if true (all devices except discs).
Word 1.(7:3) -- End of File condition:

0 -- No EDF detected.

1 -- Nardware EDF (e.g., tape nark).
2 -- :DRTA record read.
4 -- :HELLO record read.
5 -- :BYC record read.
6 -- :JOB record read.
6 -- :JOB record read.
Hord 1.(12:4) -- Device subtypes. See discussion for tape entry (below) for a description of the Ruto bit (12:1).

The remaining bits in Word 1 are device-dependent and are described with their corresponding entry diagram.

Word 1.(12:1) -- Part of Device Subtype field. If true, device is allocated automatically when opened. If false, operator must allocate.

G. 00.00 13- 6

# Entry for Disc Drives

+			11 12 13 14 15					
0  0 ////////								
Devc   J Da	NIME IRV	End of   S	F  Device					
1 Duned  o ta	SIG I	File  or	o  Subtype					
State  b	DIPVI	Cndition  F	r					
2  SYSDB-relative pointer to the DIT								
3 // 50 /////	11111111	///////////////////////////////////////	mmmmi					

Discussion (unique fields only):
Word 1.( 0:2) -- Device Dunership State. May not be 1 (owned) for shared device (system volume or private volume). Serial and foreign disce are non-sharable and nay be owned. See the full discussion of this field under Typical Entry, above.

Word 1.( 4:1) -- If true, the disc is a nonsystem donain (private volume, serial disc or foreign disc) disc drive.

Word 1.( 5:1) -- If true, disc is a mounted private volume.

Word 1.( 6:1) -- If true, the disc is a reserved volume used to satisfy the requirements of a multiple volume private volume set.

Word 1.(10:1) -- If true, the disc is a physically and logically mounted serial or foreign disc. Bits 5 and 6 must be false.

Word 1.(11:1) -- If bit 10 is true, then 1 ==> foreign disc, 0 ==> serial

Word 3.( 1:1) -- If true, the device is currently being used as a serial disc (that is, it is allocated to a user as a serial disc). This bit duplicates a bit in the LDTX entry so that this information can be found in a system (пеногу-resident) table.

G.00.00 13- 7

G.00.00

#### Logical Device Table (LDT)

#### Dverview of Data Segment

DST 14 (= X16) SIR 10 (= X12) ------DST Z16 Logical Device Table (LDT) Logical Device Table Extension (LDTX)

Logical Device Table

#### Zero Entry Format

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0	
1	Entry size = 7
2	Streams device number
3	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
4	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
5	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
6	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
7	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

G.00.00 13- 9

#### Typical Entry Format

| Yolune table index if device type = 0-7, else | | Main process pin # or spooler process pin # |1 |CS|FD| Device type Record width iz CONTROL-Y pin 15 Default output device -DR- default class index 6 (see discussion)

6.00.00 13- 10

I / 0

Logical Device Table Extension (LDTX)

#### Overview of Data Segment

DST 14 (= X16) SIR 10 (= X12) -----t---DST X16 Logical Device Table (LDT) Logical Device Table Extension (LDTX)

I / 0

Zero Entry

0	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 +
1	Entry size = 5
2	
3	
4	

#### Typical entry

0   S SD C	2 3 4 5 6 7 8 9 10 11 12 13 14 15 -             -
1	information
2	fields.
3	See the following examples
4	of LDTX entries.

#### Where:

S....Seek ahead enable/disable flag (system or PV disc only).
SD...This logical device is a Serial Disc or a Foreign Disc.
CP...This logical device uses the CIPER protocol.
FS...This is a system or PV disc with Disc Free Space management.
DS...This LDEV is a DS or data communications device.

#### Terminal Entry

				10 11 12 13 14 15
0	1000	1 01 01 01	Reserved	TBRC
1	!		Descriptor Tal	
2	IUS //	///////////////////////////////////////	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
3	1////	///////////////////////////////////////	///////////////////////////////////////	<i></i>
4	1////	111/11/11	///////////////////////////////////////	mmmi

TBRC.. Terminal's baud rate code (CPS = characters per second).

Speed (CPS) ADCC/ATP (NPIB) TBRC

Net known	0	
1920	16 (ATP only	١
960	8	•
480	ğ	
240	7	
120	11	
60	6	
30	13	
15	14	
14		
10	15	

WS....This terminal is connected to a Workstation Configurator port.

TDT offset...Offset from the base of the Terminal Descriptor Table (TDT) to the TDT entry for this terminal. A -1 indicates no TDT entry exists for this terminal.

G.00.00 13- 13

Serial or Foreign Disc Entry

+		7 8 9 10 11 12 13           erved  /////////	11+
	DISC: XDS# for DISC: 1	yarıables, Gap Tabl	e
	DISC: 1 ==> da DISC: not used	ita buffer XDS's acqu I.	ired
	DISC: PCB inde DISC: not used	x uhen WAITing, else I.	•
4 1/////	///////////////////////////////////////	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	/////

CIPER Entry

0	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 +	
1	CIPER Device Control Data Segment # (CDCDS)	i
2	DN  CTM Index for this device (CTMI)	İ
3	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	i
4	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	۱

DB....If set to 1, then debugging is in effect.
DN....If 1, the CIPER facility has been de-activated for this device because of error.
CTMI..Control Table Map Index (an index into the Control Table Nap (CTM), which is located in the CDCOS.

#### System or Private Volume Disc Entry

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0   S  0  0  1  0  Reserved  //////////
1 <b>////////////////////////////////////</b>
2   Disc Free Space DST number (DFSDST)
3   Disc Free Space error status (DFSERR)
4 /////////

S.....Seek ahead enable/disable flag.

G.00.00 13- 14

I / 0

Device Class Table (DCT)

Overview of Data Segment

DST 40 (= X50) SIR 40 (= X50) -----DST X50 Device Class Table (DCT) Terminal Descriptor Table (TDT)

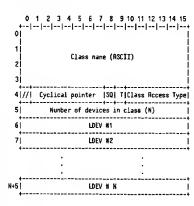
Device Class Table

Header Entry Format

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 
Entry size (variable, this word set to 1)
Number of device class entries
Pointer to first device class entry (segnent relative)
Number of terminal descriptor entries
Pointer to first terminal descriptor entry (segment relative)

I / 0

Typical Entry Format



Discussion:

The Device Class Table (DCT) contains a varying number of variable length entries. This is because you may configure an arbitrary number of device classes on a system, and each device class may be comprised of an arbitrary number of logical devices. There is one DCT entry per device class, and each DCT entry contains a list of logical devices in the class. There is no established order of entries in the DCT, nor is there an order of LDEVs within an entry.

Due to the haphazard nature of the DCT, its overall properties are kept in the header entry. These include the segment-relative starting address of the DCT (in case the header entry should be expanded later) and the number of entries in the table 8 segment-relative pointer to the Terminal Descriptor Table (which follows the DCT) may also be used to calculate the size of the DCT. Also note the "Entry size" used. It is meaningless for this table, but is included for compatibility with other fixed-length entry MPE tables. Since the DCT entries are of variable length, when you want a particular entry you must always start at the beginning of the DCT and link through each entry until you find the one you're interested in.

A few of the fields in the DCT require further description:

Word 4.( 1:7) -- Cyclical pointer. Eurrently used only for system and private volume disc devices. The pointer varies from 1 to N (number of en-

I / D

tries in the class) and indicates the LDEVM in the class list on which the last extent was allocated. The disc space allocation routines will try to satisfy the next re- quest on the next disc drive indicated by the cyclical pointer (with wraparound to 1 if the pointer > M). If that fails, the pointer is incremented until space is found or all devices in the class how been tried.

Word 4.( 8:1) -- If set, spooling has been enabled (spool queues opened) for this device class.

Word 4.( 9:1) -- If set, the class is a terminal type class.

Word 4.(10:6) -- Usually the same as the device type represented sented by the class (0 for24 for tape, 32 for printer, etc.). Serial disc classes are disc devices accessed as tape drives, so their true device types are kept in the LDT, while this field holds a special cial type (31, or 37), indicating a serial I/O (non-concurrent) device. Similarly, a foreign disc is a nonsharable disc drive, so that fact is reflected by a special type ? in this field, even though the true hard-ware type is kept in the LDT, as for serial discs.

Interrupt Linkage Table (ILT) for HP-IB Systems

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 Channel **ICPVRO** Program Variable Area (ICPVA) ICPVRO1 ICPVRO2 ICPVRO3 ICPVR04 ICPVR05 5 DMA Abort Rdd ress 6 ISRQL/ICPGM | | CHAN | DEV 7 CHANQUE ICHTRL 1 11 ISIOP %10 |\$YSDB relative pointer to channel program area.| ISTRP X11 | SYSDB relative pointer to status return area. | X12 single instruction that is executed to extract the device unit number from the status pointed to by ISTAP. X13 | SYSDB relative DIT pointer of the device | currently using the channel to perform a data ICOP operation. Z14 I SIDPSIZE COLLEN IQUEUE Z15 |RU|UP|IG|SC|SQ| X16 | SYSDB relative DIT pointer for unit 0 IDITPO |SYSDB relative DIT pointer for unit n IDITPH Program status return area pointed to by ISTAP Seeknask (Disc only)

G.00.00 13- 18

6.00.00 13- 17

I/D

#### ILT (Cont.)

IPCVR - These four words comprise the channel program variable area where information is stored concerning a channel program Interrupt instruction or abort.

ICPVRA - Words 4 and 5 contain DMR address, when channel program aborts.

ICPVRA - Words 4 and 5 contain DMR address, when channel program aborts during DMR transfer.

ISRQL - Serial poll request queue length. HP-IB Systems do not support any serial poll devices. This should always be zero.

ICPGM - This is the SYSDB relative address of the channel program to be started for this device after receiving a HIDP interrupt in GIP. GIP will call STRRID when the flags word indicates "ignore halt interrupt" and "start channel program" bits are set.

ICHTRL - Contains controller information.

If set, the controller is sharing a software channel resource in order to limit bandwidth.

CHHQ In worder for a Series 33 device is equivalent to:

CHHA - Channel rumber (4 nost significant bits of DRTH)

DEV - device number (3 least significant bits of DRTH)

DEV - device number (3 least significant bits of DRTH)

IFLAG - Used for controller flags.

RW Rumait flag. An idle channel program should be started when there are no active requests to process.

WP Lead for controller. Inthis bit is reset by an interrupt.

1.6 Ignorehi flag. An HIDP instruction has been issued against this controller, but the channel program has been started by the channel code when this program halts.

SC Start channel program "queued" flag. When bit SC is set, this bit will determine if the call to START MPIB will have logical parameter QUEUED true or false.

HCUNIT Highest configured unit number for this controller.

I / D

#### Device Information Table (DIT)

There is one DIT per physical device. If a physical device represents represents more than one logical device, the logical device number is obtained from the I/O queue element. Although details of DIT's vary with device, the following structure is common to all:

DIT for HP-IB Systems

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
O IT  D  AC RQ SI MU  O ID IA NO ST HS  STATE	DFLAG
1  SYSDB relative pointer to the DIT for the next   device requesting this resource or service	DLIHK
2  SYSDB relative pointer to the first IDQ in   request list for this device	DIDQP
3   Logical device humber	DIDEA
4 SYSDB relative pointer to Device Linkage Table	DOLTP
5  SYSDB relative pntr to Interrupt Linkage Table	DILTP
6   Controller Hardware Status	DSTAT
7 Hardware error status. Set when the driver detects an error. Whenever <>0, the driver monitor logs an I/D error and clears this word	DSERR
8   Device Dependent Area	(BTIME)
9   Device Dependent Area	(DTRQK)
10   IOT  ////////// Phys. unit H	DUNIT

Used by some device drivers, it denotes timer request index.

#### DIT Terminology for NP-IB Systems

DFLAG - DEVICE RELATIVE FLAGS
T SET IF DEVICE IS A TERMINEL.
D SET IF DEVICE IS A DISC.
AC RCIVE BIT. 1 IMPLIES A MONITOR CURRENTLY SERVICING THIS DEVICE.
RQ REDUEST BIT. 1 IMPLIES SERVICE REQUESTED WHILE PROCEEDING BIT. 1 INFLIES SERVICE REQUESTED WHILE
HONDITOR IS RETIVE.

ID IF SET, HULTIPLE UNIT CONTROLLER.

ID IF SET, THEN A CHRINEL PROGRAM IS CURRENTLY EXECUTING.

IR IF SET, HEN A CHRINEL PROGRAM IS CURRENT.

IN IF SET, DEVICE IS IN A NOT REROY OR OPERATOR WAIT.

ST IF SET, AN IOLE CHANNEL PROGRAM SHOULD BE STARTED FOR THIS DEVICE.

SI SPECIAL INTERRUPT MANDLER

NS OD NOT SHORT WHIT IT HIS OISC

STRIE CURRENT DRIVER STRIE AS DEFINED BY THE MONITOR.

RULDWARDLE STARTS ARE:

0 - STRRY REQUEST

1 - NOT USED (BUT RESERVED)

2 - CALL DRIVER INTITIOR

3 - CALL DRIVER COMPLETIOR

4 - NOT USED (BUT RESERVED)

5 - COMPLETE REDUEST

6 - UNEXPECTED INTERRUPT DECURRED

7 - START DEPARTOR INTERVENTION WAITING

X10 - WAITING (ON DEPARTOR). RESTART AT D

X11 - WAITING (ON DEPARTOR). RESTART AT D

X12 - WAITING (FOR COMPLETION INTERVENT)

X14 - WAITING (FOR COMPLETION INTERVENT)

X15 - NOT USED (BUT RESERVED)

X16 - WAITING (FOR DEVICE CONTROLLER WAILINGILITY)

X17 - WAITING (FOR DEVICE CONTROLLER WAILINGILITY)

X18 - MOT USED (BUT RESERVED)

X19 - WAITING (COMPLETION INTERVENTED THE STALL BUTTON)

X11 - WAITING (COMPLETED CODE MEKEPRESENT)

X17 - WAITING (COMPLETED CODE MEKEPRESENT)

IDT - I/D System type D-Series II/III I/D System

2-unused

3-unused

#### Device Information Table (DIT) for CIPER

There is one DIT per physical device. If a physical device represents hore than one logical device, the logical device number is obtained from the IDQ element (houever, this driver only supports one device per controller.) The following diagram shows the DIT was for the HP-IB CIPER physical driver.

3| IDT | Phys. unit # | Logical device number | DLDEA 4| SYSDB relative pointer to Device Linkage Table 5| SYSDB relative pointer to Intrp Linkage Table | DILTE 6|VS|AB|RE|TP|NR| NR CHT | DEVICE STATUS DSRVE 7 Nardware error status. Set when the driver detects an error. Whenever (>0, the driver nonitor logs an I/O error and clears this word 210 Bit D is set at completion of timer X11 Nolds the time out request entry index while a timer is active. 212[RF]UE]DE]TD|UNIT CHT|DATA CHT| TD CHT |PATY CHT| DCDUNTS Error logging location #1 Error logging location #2 DLOGCOUNT DFLAG - Flags and request state
AC ACTIVE - A nonitor is currently servicing this device.
RD AEQUEST - A service request is pending while the nonitor is REQUEST - R service request is pending while the nonitor is active.

IDPROG - Rn I/O Channel Program is running for this device.

IAK - An interrupt or response has occurred for this device.

NDTRDY - Go to state XID after Idle Channel Program is started.

STWRIT - The device nonitor is starting an Idle Channel Program for this device. There is no IDO associated with this type of request.

Attack of the device nonitor. Specifies the next action to be taken in SIDDM in servicing the request:

0 - start new request

1 - not used
2 - call driver initiator procedure
3 - call driver completor procedure
4 - not used
5 - process request completed
6 - initiate device recognition sequence
7 - start operator intervention wait

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 DI DI DIRCIRDI DI OI OI DI IRINDISTI OI STATE

1 | SYSDB relative pointer to the DIT for the next | device requesting this resource or service

2 IDD table index to the first IDD in request list for this device

DFLAG

DITHK

DIODE

G.00.00 13- 22

T / D

X10 - wait for interrupt (operator intervention)
restart at state 0
X11 - wait for data segment freeze, then state 2
X12 - wait for driver initiator to be frozen, then
allocate controller (state 2)
X13 - wait for I/O completion interrupt, then state 3
X14 - wait for controller, then call driver initiator
X15 - mot used
X16 - wait for initiator make present, then state 2
X17 - wait for completor make present, then state 3 DUDEV - I/D system type, unit and logical device number 0 - HP3000 Spries iII/III 1 - HP 3000 HP-IB 2 - Unused DSAVE - Device processing flags

VS - VALID STATUS - Set to indicate Device Status has been updated.

AB - DVRABFLAG - Sequence Rhort in progress due to a REDRI request.

FE - RETRYLAG - Sequence Rhort in progress due to an error.

TP - TIMERPOPPED - Current error is due to software tiner popping.

NR CMT - NATOMOTELAG - NAT Ready Walt in progress.

NR CMT - DEVICE STATUS - Wunder of Not Ready Walts during this request.

DEVICE STATUS - CRC available and enabled.

" 9 - Reserved.

" 10 - Reserved. Reserved.
Reserved.
Reserved.
Power fail or reset has occurred.
A protocol error has been detected.
A parity error has been detected.
The peripheral has data to send. 11 12 13 14 15 OSERR - Pointer to status to be logged.
Bits(0:8) - Number of words to be logged.
Bits(8:8) - Offset relative to DITP(0). OCCUNTS

RF - RED FAILED

UE - UNIT ERROR

DE - ORTE ERROR

TD - TIME OUT

UNIT CHT

DRTA CHT

TD CNT

PRTY CHT - Error flags and error counts (4).

Rn error has forced this request to be aborted.

The current error is a Unit Error.

The current error is a Data Error.

The current error is a Dit Time Out Error.

Hunber of Unit Errors during this request.

Number of Data Errors during this request.

Number of GIC Time Outs during this request.

Number of HP-IB Parity Errors during this request. T / D

#### DIT for Channel Devices

	D 	1	2 3	3 4	5	6	7	8	9	10	11 1	2 1	3 14	15	
D	TERMI	DISC   R	CTIRE	D  	M    TINU	SID PREM	IO  PROG	IAK	HEAD	INT IRY		S	TATE		DFLAG
1						NEXT	01TP								DLINK
2						ID	Sb.								DICCP
3					LOGIC	AL DEV	ICE N	UMBER							DLOEV
4						DL1	[P								OLTP
5	İ					ILI	P								DILTP
6				Co	ntroll	er Hai	duare	Stat	us						DSTAT
7	- <del>-</del>				Hardu	are E	ror S	tatus							DSERR
8															DTIME
g								_							DTROX
10	IDI								PN	YS.	UNIT	#			DUNIT

### DAIVER DEPENDENT DIT BREA

DFLAG.TERMINAL - Device is a terminal
DISC - Device is a Disc (Bit 0 = D)
RCTIVE - R nonitor is currently servicing this device
REDUEST - Service requested while nonitor was active

.MUNIT - device controller servicing multiple units
.SIDPREMPI- If set them a request has been queued for this device. Preempt code is set in 100.
.IOPROS - I/D program in progress. Decrement SIOCOUNT and check for nulti-channel when complete
.IRK - Interrupt or Response has occurred.
.NT RDY - Not ready for SID. SIODM holds off mext SID until RLDMPDLL is done.

DTROX - Used by some device drivers, it denotes timer request index.

#### DIT for Channel Devices (Cont.)

 $\ensuremath{\mathsf{DFLAG}}.\ensuremath{\mathsf{STATE}}$  – this quantity specifies the next action to be taken in servicing the request.

O-new - start request.
1-not used.
2-call Driver Initiator Procedure
3-call Driver Completor Procedure
5-conplete request
6-device recognition
7-start operator intervention wait (X10)
X10-restart request on interrupt
X11-wait for data to be frozen then state 2
X12-wait for driver code to be frozen then state 2
X13-call completor on interrupt
X14-wait for device controller
X15-not used
X16-wait for initiator make present then state 2
X17-wait for completor make present then state 3 0-пен - start request.

DLINK - SYSDB relative pointer to the DII for the next device requesting this resource or service.

DIDQP - SYSDB relative pointer to the first IDQ in the request list for this device requesting this resource or service.

DLOEV.LDEVN - Logical Device Number - Logical Device Comber - Logical Device Number - IDT - ID type 0-> Series ITI I/D, 1=> NPIB I/O

DDLTP - SYSDB relative pointer to the DLT.

DIITP - SYSDB relative pointer to the DLT.

DIITP - SYSDB relative pointer to the ILT.

STRIT - interrupt status for this device. Set each time the device interrupts.

DSERR - Nardware Device Controller Status. Set when the driver detects an error. Whenever not zero, SIODB logs an I/D error and clears this word.

DTINE - time out completed flags. If a timeout occurs in response to a time request type ZZO (I/O request), the sign bit is set in this word. The IR bit in DTLRG is also set, and the homitor for this device is swakened. (Dnly used if timer services are requested. Must be word H8 if timer services are requested. Nust be word H8 if timer services are requested.

G.00.00 13- 25

#### DIT For 7905/7906/7920/7925

		1			4	5	6	7	. 8	9			,12	15		
٥	0				€D	M	10	1/0	IAK	1	0	İ٥	STATE		٥	DF LAG
1	NEXT DITP													1	DLINK	
2														2	DIDQP	
3						LOG	ICA	L DEV	ICE A	PIMD	-0				ia	DLDEV
4																DDLTP
5								I	LTP						İĸ	DILTP
6							-1 I	INEN I	POWER						6	DRQST
7		# (	OF EF	AAOR	NDAC	s to	LOG		) [	ITI	REL	AOC	OR TO LO	G	7	DSEAA
8				INDE	X OF	FIRS	T AS	QUES:	T IN	QUE	JE				10	DHANG
9				INDE	X OF	LAST	AEC	UEST							11	TORRNO
10	I	T	///										MIT#		12	DUNIT
11				SID P	ROGE	AM-RE	LAT	IVE RE	ORT	ADDA	ESS	3			13	DLOGSIOP
12 13						. <b></b>	CUF	RENT DISK 1	PNYS RDDRE	ICA SS	L 				14 15	CPDA
14						CURR	ENT	DATA	BUFF	EA F	ADDE	IES:	3		16	CDBA
15						, L									17	MCR
16										COL	JT.				20	CHC
17								750HE	THO	X					21	SYSBUFA
18							SI	ATUS	1 88	THO	d			i	22	STAT1
19							<b>S</b> 1	ATUS					<del>.</del>	1	22	STATS
20								CAI							24	CEDA
21						HEAD							)R			
22			- <b>-</b>	<b>-</b>		<b>-</b> -	\$1	ATUS	1 88	TUA	(			i	i	ì
23								- 1	`YI				<b>-</b>			

G.00.00 13- 26

I / 0

DIT for 7905/7906/7920/7925 (Cont.)

	I		
24	HEAD   SECTDA		OFOUEDT
25	DISPLACEMENT		AEQUEST Syndrome
26	PATT 1		
27	PATT 2		
28	PATT 3		
29	SECTOR COUNT TO TRANSFER	35	SCOUNT
30	INITIALIZE ADDRESS	36	INITADA
31		37	
32	L	40	DMISC
33 34		41 42	SEEKSTAT
35	CPVA WORD O UPDN CNANNEL ABORT	43	DLOGERADA
36	CUAAENT LOGICAL SECTOR AODAESS	44	CLDR
į			
!			

DMISC (15:1) L'STAT'EAA - 1 Last transfer ended in error.

IDT - I/D Devices 0 - non-HP-IB 1 - NP-IB Systems 2 - unused 3 - unused

I / 0

Error and Retry Information

													13 14 15			
1	-	- <b>-</b>		<b>-</b> -					<b> </b>	l		I				
11	)  3	E	l M	l li	T	0	C	CL	0	0	0	٥	retry ont	QMISC	OF	1
1-	.	1	I	II		11	11	l	I				II			

D - retry determination
S - request syndrome
E - request error information
M - update track map
U - writing track map
C - issued a recalibration
CL driver issuing channel clear
I - timeout wait

NDTE: Integrated Cartridge Tape's DIT has the same format.

#### CS 80 Disc Device Information Table (OIT)

There is one DIT per physical device. If a physical device represents nore than one logical device, the logical device number is obtained from the CS'80 disc controller, there will only be one device. The following diagram shows the DIT used by the CS'80 disc driver.

0 1 2 3 4 5 6 7 B 9 10 11 12 13 14 15 MMEMDNIC

1| SYSDB relative pointer to the DIT for the next| OLIHK | device requesting

21	Current request index	DCURREQP
31	Logical device number	DLDEV

4| SYSDB relative pointer to Device Linkage Table| DOLTP

5| SYSDB relative pointer to Intrp Linkage Table | DILTP

6| OSTRT is -1 when a system powerfail occurred | DSTRT

7| Nardware error status. Set when the driver | DSERR | detects an error. Uhanaver < XO. the driver | | monitor logs an I/O error and clears this

Horo	never <>0, the driver     Honitor logs an ] i	:/O error a
X10	index of firet request in queue	DQNERO *
<b>Z11</b>	index of last request in queue	DQTRIL *
212	IOT   Physical Unit #	DUNIT
<b>%13</b>	Table relative index to system buffer element	DSBUFROOR
<b>X14</b>	High order logical sector address of bad blk	DBRDBLK1
<b>X</b> 15	Low order logical sector address of bad blk	DBRDBLK2
<b>X16</b>	Byte transfer left when bad block occurred	DBROXFER
<b>X17</b>	Mardware logged error status - CPVR (0)	DLOGERROR
X 20 j	Channel program aborted relative offset	DSIOPSTOP
X21	Oisc status (20 bytes)-Logged on status error	DSTRTUS
٠į		
·į		
233	LK   IF   MO   SUBSTRTE	OMISC

6.00.00 13- 29

X34 | RE | DC | DR | EN | |LOCKL STRTE| RPSWORO1 TŽ % 35 J | RPSWORO2

DFLAG - Flags and request state

IN TERM — Set if device is a terminal. DS DISC — If TM = 0 and this bit is set then the device is a disc, otherwise device dependent. RC RCTIVE — R monitor is currently servicing this device. RQ REQUEST — R service request is pending while the monitor is active. IO IOPROG — RR I/O Channel Program is running for this device. IR IRK — RR interrupt or response has occurred for this device. HO NOTROY — Go to state 2/O after Idle Channel Program is started. ST STWRIT — The device monitor is starting an Idle Channel Program for this device. There is no IQQ associated with this type of request. STRTE — State of the device monitor. Specifies the next action to be taken in SIODH in servicing the request:

0 - start new request 1 - not used 2 - call driver initiator procedure 3 - call driver completor procedure 4 - not used 5 - process request completed 6 - initiate device recognition sequence 7 - start operator intervention wall 70 - wait for interrupt (operator intervention) restart at state 0 X11 - wait for data segment freeze, then state 2 X12 - wait for driver initiator to be frozen, then allocate controller (state 2) X13 - wait for I/O completion interrupt, then state 3 X14 - wait for controller, then call driver initiator X15 - not used X16 - wait for initiator make present, then state 2 X17 - wait for completor make present, then state 2 X17 - wait

DLINK - R SYSDB relative pointer to the next OIT requesting this resource or

DCURREQP - R current request sysbase index.

OUNIT.(0:2) - I/D system type

0 - non-HP-IB 1 - HP3000 HP-IB Systems 2 - Unused 3 - Unused

DLDEV - Logical device number of this device.

DSTRT - Set to a -1 when a system powerfail has occurred.

DSERR - Pointer to status to be logged.

Bits(0:7) - Humber of words to be logged. Bits(B:15) - Offset relative to  $\mathtt{DITP}(0)$ .

DMISC - Device dependent processing flags

 $\ensuremath{\mathsf{LOCK}}\xspace^* \ensuremath{\mathsf{FLG}}\xspace - \ensuremath{\mathsf{Lock}}\xspace \ensuremath{\mathsf{flag}}\xspace \ensuremath{\mathsf{denoting}}\xspace \ensuremath{\mathsf{unload}}\xspace \ensuremath{\mathsf{status}}\xspace \ensuremath{\mathsf{of}}\xspace \ensuremath{\mathsf{the}}\xspace \ensuremath{\mathsf{disc}}\xspace \ensuremath{\mathsf{volume}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{otd}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace \ensuremath{\mathsf{e}}\xspace$ 

0 - RIlow operator unload to the volume. 1 - Oeny operator unload to the

G.00.00 13- 30

I/D

IGHORE'INT'FLG - Ignore unexpected interrupt flag.

SUBSTRIE - Indicates state of the idle channel program:

O - Mormal idle channel program wait 1 - Idle request being serviced wait

 $\ensuremath{\mathsf{DSBUFRDDR}}$  – SYSDB relative pointer to the system buffer element used to read the DSCT. Zero, if no element gotten.

DBRDBLK1 - High order logical sector address of the bad block for the Defective Sector Table (DSCT) entry.

DBROBLK2 - Low order logical sector address of the bad block for the DSCT

DBROXFER - Byte transfer left when bad block occurred.

DLOGERROR - CPVR(0) logged on hardware error status.

<code>OSIOPSTOP</code> - Stopped channel program relative offset location due to an error in  $\mbox{CPVR}(0)$ .

DSTRTUS - 20 bytes disc status logged on status error. (See CS'80 Disc Drive

RPSWORD1 - Flags and local state

RE - Read revision code done. Set if read revision code level is done. DC - RPS revision code. Set if controller is "PEP"ed. DR - RPS desirable. Set if RPS is desirable. EH - RPS enabled. Set if default value for RPS is enabled. NR - Driver is processing a harginal data error fron the drive. Do not return hard error. Local State - State of the local request hade by

 ${\tt O}$  -  ${\tt No}$  local request is being processed 1 - Reading rev code 2 - Setting default RPS

RPSWDRD2 - Default value for RPS

 $\ensuremath{\mathsf{T1}}$  – Time to target in hundreds of microseconds  $\ensuremath{\mathsf{T2}}$  –  $\ensuremath{\mathsf{Window}}$  size in hundreds of microseconds

I / 0

### OIT For 7970 Magnetic Tape

	0 1 2 3 4 5 6 7 8 9 10 11 12 15 0 D RCT REQ 0 M 0 L/O IRK 0 0 0 STRTE	DFLAG
1	HEXT DITP	DLINK
2	IDQP	DIOQP
3	LOGICRL DEVICE NUMBER	DLOEA
4	DLT PTR	DOLTP
5	ILT PTR	DILTP
6	RN RU SH CE DC  HRRDNRRE STRTUS	DSTRT
7	ERROR STRTUS	DSERR
8	TIMEOUT FLAGS	DTIME
9	TIMER REQUEST INDEX	DTRQX
10	IDT  ////////## PMYSICRL UNIT #	DUHIT
11	13 R84  RW	DOFLEGS

IOT - I/O Devices

0 - non-HP-IB 1 - HP-IB Systems 3 - unused 4 - unused

DSRVE — Device processing flags
RN RWBIT — Indicates tape has been remound.
RU RWUNLO — Indicates that a rewind/unload was performed to allow a write-ring mount.
SN SNORT — R short read is in progress. Rfter completion of read, EOF is checked for and if not present, the requested bytes are transferred from the short-read buffer to the user's buffer.

CE CESTAT — Channel parity error processing is in progress.
DC DSFIRG — Transfer used data chaining — used for computing the transmission log.
RU — (ODFIRGS, bit 15) if set, tape is remound
RB4 — (bit 14) if set, need to reuind tape before next write

#### OMISC

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

FORMARD  BRCK
I RI BI FI GI EI SI UI SPRCE   SPRCE   RETRY

- Where
  R retry in progress
  B backspace in progress
  F formard space in progress
  G gap in progress
  E backspace on data end-of-file
  S short read in progress
  U unload tape for write ring installation

G.00.00 13- 33

#### DIT for 7976 Magnetic Tape

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element. The following diagram shows the DIT used for the mag tape driver.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEMONIC
O  O  O RC RQ  O MU  O IO IA  O  O  O  STRTE	DFLAG
i  SYSOB relative pointer to the DIT for the next   device requesting this resource or service	DLINK
2  SYSOB relative pointer to the first IOQ in   request list for this device	OIOQP
3  Logical device number	DLDEA
4  SYSOB relative pointer to Oevice Linkage Table	00 LTP
5  SYSOB relative potr to Interrupt Linkage Table	OILTP
6 RW RW SH   DC PF	DSRVE
7  Hardware error status. Set when the driver   detects an error. Whenever <>0, the driver   nonitor logs an I/O error and clears this word	DSERR
X10   Bit 0 is set at completion of timer	DTIME
X11   Interrupt status for this unit. Set by the   driver each time it processes an interrupt.	DSTRT
%12  IDT  /////////// Physical unit #	
X13 Holds the time out request entry index while a timer is active.	ORQST
214 Error log. Contains 5 valid bytes of status	OLOGERROR

DFLAG - Flags and request state
RC ACTIVE - R nonitor is currently servicing this device.
RO REDUEST - A service request is pending while the monitor is

REDUEST - A service request is pending while the monitor is active.

HUNTI - This device is on a nulti-unit controller.

10PR06 - Rn I/O Channel Program is running for this device.

RR R R interrupt or response has occurred for this device.

NOTRDY - Go to state X10 after Idle Channel Program is started.

- The device nonitor is starting an Idle Channel Program for this device. There is no IQQ associated with this type of request.

G.00.00 13- 34

I / 0

STATE

- State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:

0 - start new request
1 - not used
2 - call driver initiator procedure
3 - call driver completor procedure
4 - not used
5 - process request completed
6 - initiate device recognition sequence
7 - start operator intervention wait
X10 - wait for interrupt (operator intervention)
restart at state 0
X11 - wait for data segment freeze, then state 2
X12 - wait for driver initiator to be frozen, then
allocate controller (state 2)
X13 - wait for I/O completion interrupt, then state 3
X14 - wait for I/O completion interrupt, then state 3
X15 - mot used
X16 - wait for initiator make present, then state 2
X17 - wait for completor make present, then state 3

OSRVE - Oevice processing flags
RW RWBIT - Indicates tape has been rewound.
RW RWBID - Indicates that a rewind/unload was performed to allow a write-ring mount.
SH SHORT - A short read is in progress. After completion of read, EUF is checked for and if not present, the requested bytes are transferred fron the short-read buffer to the user's buffer.

G.00.00 13- 35

DC DSFLRG – Transfer used data chaining – used for computing the transmission log. PF POWER – Device power up indication.

I / 0

OSTRT - Mag tape controller status

BITS USE

0 END OF FILE (EOF)

BEGINNING OF TRPE (BOT) / LORD POINT (LP) ENO OF TRPE (EOT) SINGLE TRACK ERROR (NOT LOGGED FOR REROS)

COMMRNO REJECT (REJECT)
FILE PROTECT (NOT WRITE EMBBLEO; NO WRITE RING)
MULTIPLE TRACK ERROR (MTE) 4 5 6

UNIT DNLINE GCR (6250 BPI DENSITY) UNIT NUMBER (MSB)

UNIT HUMBER (£SB) TIMING ERROR TAPE RUNRWAY

10 11 12

REWINDING \* \* (REPORTED AS UNIT NOT READY) INTERFRCE BUSY \*

#### Card Reader DIT

	O 1 2 3 4 5 6 7 8 9 10 11 12 15 O 0 ACT RED O O I I/O I IAK READ IN RET INSTATE	DFLAG
1	OTLE TINK ID HEXT DIT	DLINK
2	IODP POINTER TO 1st REDUEST	OIGQP
3	LOGICAL DEVICE NUMBER	DLDEV
4	DRIVER LINKAGE TABLE POINTER	DDLTP
5	INTERRUPT LINKAGE TABLE POINTER	DILTP
6	(SEE BELOW)	OSTAT
7	ERRDR STATUS IF NOT O	DSEAR
210	REDUESTED ADRD COUNT	OTIME
<b>211</b>	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	DTRDX
212	IOT  ////////#############################	DUNIT

DSTAT bits:

BIT15=NOT REBOY

BITO=SID UN
BIT1=0
BIT2=INT PENDING
BIT3=ITITUG ERROR
BIT4=LIGHT DARK CNECK
BIT5 5-6 = 00 CDLUMN BINARY MODE
01 UNUSED
10 PRICKED BINARY MODE
11 HOLLERITH-TO-ASCII MODE BITO-SID DK BIT8=EDF DETECTED
BITS 9-10 = OO HORMAL
O1 HOPPER EMPTY
10 UNUSED 11 STACKER FULL BIT11=INVALID HOLLERITH
BIT12=PICK FAIL OR HOTOR CHECK
BIT13=TEST
BIT14=TROUBLE

G.00.00 13- 37

#### Card Reader DIT Field Omfinitions

DFLRG - Flags and device state

RCTIVE Monitor is currently active servicing this device.

REDUEST Service for this device was requested while the monitor was active.

TOPROG SIO program in progress.

IAK Interrupt occurred or request aborted or preenpted.

Previous read resulted in an EOF with a backup save requested. The data has been saved in an auxiliary buffer and will be passed back on the next read requ READDONE

Set when a not ready message has been issued, and cleared when the reader is found ready. Used to prevent multiple Not Ready messages when power is turned on. NRMESSAGE

MSTATE Monitor State. See SIDDM specifications for details.

DLINK - SYSDB relative pointer to the DIT for the next device requesting service for this resource.

DIODP - SYSOB relative pointer to the first IDD element in the request list for this device.

DLDEV - Logical device number and unit number.

INTT Unit number of device. Logical device number.

OOLTP - SYSDB relative pointer to driver linkage table (DLT).

DSTAT - Device interrupt status. Contains the device interrupt status at the last interrupt. See hardware ERS for details.

DSERR - Device interrupt error status. If not zero, then holds the device interrupt status from an operation with an erroneous completion status. Causes SIDDM to log an error.

DUCNT - Nolds the requested transfer count in words.

G.00.00 13- 38

I/D

### Device Information Table for HP-IB Card Reader

There is one OIT per physical device. If a physical device represents nore than one logical device, the logical device number is obtained from the IDD element. The following diagram shows the OIT used for the card reader driver.

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEMONIC			
٥	O O O   AC   RD   O   MU   O   I O   I O   O   STATE	DFLAG			
1	SYSDB relative pointer to the DIT for the next device requesting this resource or service	DLINK			
2	IOD table relative index to the first IDD in request list for this device	DIDDP			
3	Logical device number	DLDEV			
4	SYSDB relative pointer to Device Linkage Table	COLTP			
5	SYSDB relative pntr to Interrupt Linkage Table	OILTP			
6	AD AF	DSRVE			
7	Nardware error status. Set when the driver detects an error. Whenever <no, an="" and="" clears="" driver="" error="" i="" logs="" nonitor="" o="" td="" the="" this="" word<=""><td>DSERR</td></no,>	DSERR			
	Not Used	DTIME			
211	Request word count	OHENT			
	IOT  ////////// Physical unit #	DUNIT			
<b>Z13</b>	Device Status. Read from device during each execution of the channel program.	DSTAT			
214	logging will be done from here.	DLOGERROR			
DFURG - Flags and request state  AC RCTIVE - R nonitor is currently servicing this device.  RD REQUEST - R service request is pending while the nonitor is active.  MU MINIT - This device is on a nulti-unit controller.  ID IDPROG - Rn I/O Channel Program is running for this device.  IA IRK - Rn interrupt or response has occurred for this device.  NO NOTROY - Go to state I/O after Idle Channel Program is started.  ST STAPIT - The device Honitor is starting an Idle Channel Program for this device. There is no 100 associated with this type of request.					

I / D

STATE

- State of the device monitor. Specifies the next action to be taken in \$100M in servicing the request:

0 - start new request
1 - not used
2 - call driver initiator procedure
3 - call driver completor procedure
4 - not used

4 - not used
5 - process request completed
6 - initiate device recognition sequence
7 - start operator intervention wait
10 - wait for interrupt (operator intervention)
restart at state 0
11 - wait for data segment freeze, then state 2
12 - wait for driver initiator to be frozen, then
allocate controller (state 2)
13 - wait for I/O completion interrupt, then state 3
14 - wait for controller, then call driver initiator
15 - not used
16 - wait for initiator nake present, then state 2
17 - wait for completor nake present, then state 3

DLDEY - Device logical device number

IDT I/D TYPE - I/D System type

0 = Series II / III I/O system

1 = HP-IB Systems

DSRVE - Device processing flags RD READDONE - R card has already been read. RF ABDRTFLRG - A device clear has already been sent for this series of aborted IDDs.

#### 2608 Line Printer DIT (NP-IB Systems)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IDQ element (however, there is only one device per 2608 controller.) The following diagram shows the DIT used for the 2608 line printer driver.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEMDNIC
O O O O O O O O O O O O O O O O O O O	DF LRG
1  SYSOB relative pointer to the DIT for the next   device requesting this resource or service	DLINK
2  IDQ table relative index to the first IDQ in   request list for this device	DIOQP
3 Logical device number	DLDEV
4  SYSDB relative pointer to Device Linkage Table	DDLTP
5  SYSDB relative pntr to Interrupt Linkage Table	DILTP
6 VM    TRB    PS FL TP	DSAVE
7) Hardware error pointer. Set when the driver detects an error. Whenever <>0, the driver nonitor logs an I/D error and clears this word	DSERR
X10  Bit 0 is set at completion of timer	DTIME
X11 Nolds the time out request entry index while a timer is active.	DRQST
%12   IOT  //////////   Physical Unit #	OUNIT
X13  Narduare logged error status	DLOGERROR

- OFLAG flags and request state
  RC RCTIVE R monitor is currently servicing this device.
  RQ REQUEST R service request is pending while the monitor is
- ID IOPROG Rn I/O Channel Program is running for this device.

  RI IRK Rn interrupt or response has occurred for this device.

  ND NOTRDY Go to state X10 after Idle Channel Program is started.

  ST STURIT The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.

G.00.∞ 13- 41

STRTE

- State of the device monitor. Specifies the next action to be taken in SIDDM in servicing the request:

  0 start new request
  1 not used
  2 call driver initiator procedure
  3 call driver completor procedure
  4 not used
  5 process nowings completed

- 4 not used
  5 process request completed
  6 initiate device recognition sequence
  7 start operator intervention wait
  X10 wait for interrupt (operator intervention)
  restart at state 0
  X11 wait for data segment freeze, then state 2
  X12 wait for data segment freeze, then state 2
  X13 wait for or initiator to be frozen, then
  allocate controller (state 2)
  X13 wait for I/0 conpletion interrupt, then state 3
  X14 wait for I/0 conpletion interrupt, initiator
  X15 not used
  X16 wait for initiator make present, then state 2
  X17 wait for completion make present, then state 3
- OLDEV I/O system type, unit and logical device number IDT I/O TYPE- Type of I/O system

  O HP3000 Series II/III

  1 NP3000 NP-IB Systems

  2 unused

  3 unused

- DSRVE Device processing flags
  VM VFCHOD VFC has been modified.
  TRB TABPFRUIT System tab default.
  PS PRESPRIE Last request used prespacing.
  FL FULL Line printer buffer is full.
  TP TOP Printer is at top of form

6.00.00 13- 42

I / 0

2608 Line Printer Status

BYTE 1 & BYTE 2: BITS USE

- 0 ON LINE
- NOT READY VFC CHANNEL 9 (BOTTOM OF FORM) VFC CHANNEL 12 (TOP OF FORM)
- VEC THITTELIZED 6/8 LINES PER INCH (NOT USED)
- POWER RESTORED/UNIT RESET
- ON LINE PRINT MECH ERROR
- SELF TEST FRILURE PRPER ERROR SELF TEST MODE 11 12
- 13
- 6/8 LPI PLATEN/RIBBON ERROR (NDT USED) 14 15

- 15 (NOT USED)

  BYTE 3: PRIMT HODE
  BITS 0-7 NODE NUMBER

  BYTE 4: PRIMRRY/SECONDRRY
  BITS 0-3 SECONDRRY CHRRCTER SET CODE
  BYTE 5: SELF TEST
  BITS 0-7 PRIST NUMBER

  BYTE 6: 6 LPI DOT ROU COUNT
  BYTE 7: 6 LPI FORN LINE HUMBER
  BYTE 8: 6 LPI FORN LINE HUMBER
  BYTE 8: 6 LPI FORN LINE HUMBER
  BYTE 9: 8 LPI FORN LINE HUMBER
  BYTE 10: 8 LPI FORN LINE HUMBER
  BYTE 10: 8 LPI FORN LINE HUMBER
  BYTE 11: FIRNINGER TOENTIFICITION CODE
  BYTE 12: FIRNINGE TOENTIFICITION CODE
  BYTE 10: 8 LPI FORN LENGTH IN LINES
  BYTE 12: FIRNINGE TOENTIFICITION CODE
  BYTE 3: 6 PODER-UP LENGUISE
  BYTE 3: 6 PODER-UP LENGUISE
  BYTE 3: 6 PODER-UP LENGUISE
  BYTE 3: 6 PODER-UP LENGUISE
  BYTE 3: 6 PODER-UP LENGUISE

I / 0

HP 2619R or 2613 Line Printer DIT (NP-IB Systems)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element (housever, there is only one device per 2631 controller.) The following diagram shows the DIT used for the 2631 line printer driver.

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEMONIC
0	O O O O RC   ROL O   O   O   O   I O   I R   NO   ST   O   STATE	DF LAG
1	SYSDB relative pointer to the DII for the next device requesting this resource or service	DLINK
2	IOQ table relative index to the first IOQ in request list for this device	DIOQP
3	Logical device number	DLDEV
4	SYSDB relative pointer to Device Linkage Table	DDLTP
5	SYSDB relative pntr to Interrupt Linkage Table	OILTP
6	BJ AB PS FL TP	DSAVE
7	Hardware error status. Set when the driver detects an error. Whenever <>0, the driver monitor logs an I/O error and clears this word	DSERR
Z 10	Bit 0 is set at completion of timer	DTIME
Z11	Holds the time out request entry index while a timer is active.	DRQST
Z12	IOT  ///////////// Physical unit #	DUNIT
z13	Nardware logged error status	DLOGERROR

- DFLAG Flags and request state
  RC ACTIVE R monitor is currently servicing this device.
  RQ REQUEST R service request is pending while the monitor is

- TO TOPROS Rn I/O Channel Program is running for this device.

  NO NOTRDY Co to state %10 after Idle Channel Program is started.

  ST STURIT The device monitor is starting an Idle Channel Program for this device. There is no IDQ associated with this type of request.

I / 0

HP 2680R/2688R 0IT

```
- State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:

O - start new request

1 - not used
2 - call driver initiator procedure
3 - call driver completor procedure
4 - not used
5 - process request completed
6 - initiate device recognition sequence
7 - start operator intervention wait
110 - wait for interrupt (operator intervention)
restart at state D
111 - wait for data segment freeze, then state 2
212 - wait for driver initiator to be frozen, then
allocate controller (state 2)
213 - wait for I/O completion interrupt, then state 3
214 - wait for controller, then call driver initiator
115 - not used
216 - wait for initiator make present, then state 2
217 - wait for completon make present, then state 3
              STRIE
              DEV - I/O system type, unit and logical device number
IOI I/O TYPE - Type of I/O system
0 - HP9500 Series 2/3
1 - HP9300 HP-IB Systems
DSRVE - Device processing flags
BJ BETJOB - Between jobs flag. If set, suppress
Powerfall nessage.
RB ABDRT - Retro (caused by Powerfail or Operator)
                                     PRESPACE - Last request used prespacing.
FULL - Line printer buffer is full.
TOP - Printer is at top of form
```

G.00.00 13- 45

IF SET, DEVICE IS IN R NOT READY OR OPERATOR WAIT.

IF SET, AN IDLE CHANNEL PROGRAM SHOULD BE STARTED
FOR THIS DEVICE.

CURRENT DRIVER STATE AS DEFINED BY THE MONITOR.

RILDWANDLE STATES ARE:

D - START REQUEST
1 - NOT USED/BUT RESERVED)
2 - CALL DRIVER INITIATIOR
3 - CALL DRIVER COMPLETOR
4 - UNISCO/BUT RESERVED)
5 - COMPLETE REQUEST. PERHAPS RETURN TO USER.
6 - UNEXPOCITED INTERMEDT OCCURRED.
7 - START OPERATOR INTERVENTION WAIT.
210 - MAITING (ON DEPERATOR). RESTART AT O.

11 - WAITING (ON DEPERATOR). RESTART AT O.

12 - WAITING (STANDAM STANDAM SHEEPESENT/FREEZE).

13 - WAITING (FOR COMPLETION INTERVENT).

14 - WAITING (FOR COMPLETION INTERVENT).

15 - UNUSED/BUT RESERVED).

16 - WAITING (CINCIPATION INTERVENT).

17 - WAITING (CONTELLING CODE MAKEPRESENT).

VESTEM TYPE INIT AND LOGICAL DEVICE NUMBER. **HSTRTE** DLDEV - I/O SYSTEM TYPE, UNIT AMO LOGICAL DEVICE MUMBER.

IOT I/O SYSTEM TYPE.

0 - HP3000 SERIES II/III (SID/0IO)
1 - HP-IB Systems
2 - RESERVED DCBCHT - CURRENT BYTE COUNT TO BE TRANSFERRED. OCUCHT - CURRENT WORD COUNT TO BE TRANSFERRED. ORCHT - REMRINING WORD COUNT TO TRANSFER. OOFFSET - OFFSET IN BUFFER OF NEXT # HORDS TO TRANSFER. DOEBUG - IF BIT 15=1 THEN DEBUGGING INFO WILL BE SENT TO CONSOLE OLOGBUFFER - STRTUS WORDS 1 & 3 RRE MOVED HERE TO BE LDGGED IF THEY WERE LOGGED FROM THE I/O STRTUS BLOCK THEIR CONTENTS MIGHT BE CHANGED BEFORE THEY WERE LDGGEO. DIOSTRY - I/O STRTUS RRER 16 HORDS, SEE I/O STRTUS BLOCK DEFINITION.

D 1 2 3 4 5 6 7 B 9 10 11 12 13 14 15 OITD !O !O !AC!RO!O !O !SP!CP!IR!NR!SN! ! STRTE ! DFLAG POINTER TO NEXT OIT OLINK 2 INDEX TO ACTIVE IOQ OR ZERO OIOOP 3 LOGICAL DEVICE NUMBER OFDEA DRIVER LINKAGE TABLE POINTER DOI TP 5 ! INTERRUPT LINKAGE TRBLE POINTER DILTP SPECIAL ERROR CONDITIONS TO BE LOGGED 6 DSTRT 7 ! ERROR LOGGING INFORMATION ! OSERR 8 !T! TIMEOUT INDICATION IN BIT O TIMER REQUEST INDEK (TRL) OR ZERO 9 ! DTRLX ! IOT !/////////! PHYSICAL UNIT N ! DUNIT 1D CURRENT DATE WRITE BYTE COUNT 11 DCBCNT CURRENT DATE WORD COUNT ! OCUCHT N OF WORDS LEFT TO TRANSFER 13 ! ! ORCHT 14 ! BUFFER OFFSET FOR NEKT N DF WORDS TO KFER. ! DOFFSET ID! DOEBUG 16 ! I/O STATUS BLOCK WORD 1 GETS LOGGEO FROM HERE ! DLOGBUFFER 17 ! I/O STATUS BLOCK WORD 3 GETS LOGGED FROM HERE ! 18/33 ! I/D STATUS ARER (16 HORDS, SEE DEFINITION) ! OIOSTAT OFLAG - DEVICE RELATIVE FLAGS.

AC RCTIVE BIT. 1 IMPLIES R MONITOR CURRENTLY
SERVICING THIS DEVICE.

RQ REQUEST BIT. 1 IMPLIES SERVICE REQUESTED
HILLE HONITOR IS ACTIVE.

SP SID PREEMPTION. IF SET THEN A PREEMPTIVE
REQUEST HAS BEEN QUEUED FOR THIS DEVICE.
PREEMPT CODE IS SET IN IOO FLERENT.

CP CHANNEL PROGRAM IN PROGRESS. IF SET, THEN
R CHANNEL PROGRAM IS CURRENTLY EXECUTING.

IR IF SET, AN INTERRUPT OR RESPONSE HAS OCCURRED.

I/O Status Block 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 10 !--THE "OR" OF HORDS 1/15 IS LOCRTED HERE----! DIT 17 ! ! ! ! (RESERVED) ! ! ! ! ! 19 3 Ī NCS FRULT NUMBER 20 4 !CL!FL!VL!CU!FU!VU!IL!IP!ST!SB!IR!MP!NJ!NM!TL!NC! 22 23 25 27 28 RECORD NUMBER OF ERROR 12 29 IF HORD 4 IS NON-ZERD 14 ! SHEET HUMBER OF ERROR IF WORD 4 IS NOH-ZERO ! 31 DR 15 ! LAST SHEET TRANSFERRED IF "JOB" & POWER-ON 32

WORD D - ERCH BIT IS THE 'DR' OF ONE WORD IN THE TRBLE (EXCEPT BIT O WHICH IS HOT USED). THEREFORE, BIT .(1:1) IS SET IF WORD 1 IN THE TRBLE IS NON-ZERO.

UORO 1 - BIT= 0 - (DF) OHLINE/DFFLINE BIT.

1 - (NS) NESSAGE BEING DISPLIYED ON THE 2680R/2688R CONSOLE.
2 - (PU) POWER WE FOODSPLETED SINCE LAST I/O STATUS RERD.
3 - (PE) PARTIY ERROR DETECTED ON PAI COMMRNO.
4 - (TE) TEANSNISSION ERROR DETECTED IN THE PRINTER.
5/15 - RESERVED. UNUSED.

NORO 2 - NOT USED. RESERVEO.

MORD 3 - HCS FRULT NUMBER. CONTAINS AN INTEGER DESCRIBING THE LAST FRULT TO OCCUR SINCE THE LAST TIME THE 1/D STRTUS WRS RENO OR THE HP 2680R/2688R WRS POWEREO DOWN. IF THE WORD IS ZERO THERE

IS HD DCS FAULT. SEE DCS ERS FOR A DESCRIPTION OF THE DCS FAULT HUMBERS.

WORO 4 - BIT= D

ULT HUMBERS.

T= D - (CL) HO ROOM FOR RITEMPTED CHARACTER SET LORO.

1 - (FL) NO ROOM FOR RITEMPTED CHARACTER SET LORO.

2 - (VL) HO ROOM FOR RITEMPTED FORM LORD.

2 - (VL) HO ROOM FOR RITEMPTED VFC LORD.

3 - (CU) HITEMPT TO PRINT DATA RHO THERE IS NO CURRENTLY SELECTED CHARACTER SET.

4 - (FU) RITEMPT TO SELECT BY UNDEFINED FORM SET.

5 - (VU) RITEMPT TO SELECT BY UNDEFINED FORM SET.

6 - (IL) RITEMPT TO PRINT DATA RHO THERE IS NO CURRENTLY SELECTED VFC SET.

6 - (IL) RITEMPT TO PRINT DATA RHO THERE IS NO CURRENTLY SELECTED LOGICAL PROGE TRABLE (LPT) ENTRY.

7 - (IP) RITEMPT TO MOVE PEH OFF THE LOGICAL PROGE.

8 - (ST) THE 2680R/2683R COULD NOT PROCESS ALL OF THE DATA BEFORE IT HAS SUPPOSEO TO BE TRANSFERRED TO THE ORUM/PAPER. ORTH HAS LOST!

9 - (SB) SPOOLER BLOCK CONTRINS FORMST ERROR.

10 - (IR) INVALID RECOVERY BLOCK RECEIVED FROM SPOOLER.

11 - (IP) INVALID NECOVERY BLOCK RECEIVED FROM SPOOLER.

12 - (NL) R COMINNO OR FUNCTION CODE HAS RECEIVED WHEN NO "JOB" HIS IN PROCESS SETTING THE HAXIMUM COPIES PER PRESENTED FUNCTION HAS INFORCESS. THE COMINNO OR FUNCTION HAS INFORCESS.

13 - (NT) NO MEMORY. 2680R/2683R DYNRHIC MEMORY ALLOCATION HAS DETECTED THAT HAIN MEMORY IS COMPLETELY OCCUPIED WITH CHARRETER SETS, VEC'S, FORMS ROM DATA SUCH THAT THE 2680R/2683R CHAMOT PROCESS THE CURRENT INPUT DATA. DATA WILL BE LOST!

14 - (IL) RITEMPT TO PRINT DATA AND THERE ARE MORE THAN HE MAXIMUM RULDWABLE LOGICAL PROFE TRABE (LPT)

ETHERS SELECTED.

5 - (NC) R NOM-EXISTENT VFC CHANNEL WAS SKIPPED TO.

NORO 5 - BIT= D - (LP) LOGICAL PRGE TRUNCRTED TO FIT PHYSICAL PRGE.
1 - (PF) PRGE SIZE REQUIRED BY PROGRAMMER OID NOT MATCH PRGE SIZE SET BY OPERATOR. OPERATOR PRGE SIZE PREVAILS

2 - (NC) NO CHRRACTER SET SELECTED.

WORDS 6/11 NOT USED BUT RESERVED FOR FUTURE USE.

WORDS 12/13 - THE RECORD NUMBER UHICH COHTRIHS THE OFFENDING ERROR
AS DEFINED BY WORD FOUN. IF A POWER FAIL DCCURS DURING
A "JOB". THE POWER FAIL BIT IS SET RHO A SHEET HUMBER IS
RHOE RVAILABLE IN WORDS FOWRIEN AND FIFTEEN. HONEVER,
THE RECORD HUMBER IS LOST AHD CHNNOT BE REPORTED. THESE
MORDS DCCUR IN A "JOB" ONLY.

HORDS 14/15 - THE SHEET HUMBER OH WHICH THE ERROR OCCURRED AS DEFINED BY HORD FOUR. IF AN ERROR OCCURS IN THE ENVIRONMENT FILE AT THE STRRT OF A "JOB", THEN THIS NUMBER WILL BE ZERO.

IH RODITIOH, WHEH A POWER FAIL OCCURS OURING R "JOB", THE POWER ON BIT IS SET IN WORD ONE AND THE SWEET HUMBER OF THE LAST SUCCESSFULLY TRANSFERRED PROEC IS PLACED HERE. THIS INFORMATION IS FOR USE BY THE SPODICE SHOULD A RECOVERY OF A "JOB" BE DETERMINED. THESE WORDS OCCUR IH "JOB" ONLY.

RLL WORDS DF THE I/D STATUS ARE CLEARED WHENEVER THE STATUS BLDCK IS RETURNED TO THE HOST. IT IS UP TO THE HOST CPU TO RETAIN ANY ONGOING STATUS BITS REQUIRED.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1003 !MB!RB!RB!ID!TO! ! XFER ! PARITY ! ! OHISC

WHERE:

USER REQUESTED TRANSFER IN EXCESS OF 4096 NORDS. THE DRIVER CAN WRITE UP TO 4096 NORDS TO THE 2680A/2688A. IN DRIVER TO HANDLE UP TO 32K NORDS, MILITUPLE WRITES REGUESD WITHOUT R RETURN TO THE USER WHO CRILED THE DRIVER. THIS BIT INDICATES THAT MULTIPLE WRITES ARE BEING DONE TO THE 2680A/2688A. .(0:1) - MB

.(1:1) - RB THE CURRENT WRITE BLOCK MUST BE RETRIED.

.(2:1) - AB USER REQUESTED RBORT IN PROGRESS FLAG.

.(3:1) - I0I/O STATUS HAS BEEN READ AND IS AVAILABLE.

.(4:1) - TO GENERAL I/O CONTROLLER TIMED OUT.

.(5:4) - RESERVED NOT CURRENTLY USEO.

.(9:3) - XFER 2680R/268BR TRRNSFER ERROR COUNTER.

.(12:3)- PARITY CHANNEL PROGRAM COMMAND PARITY ERROR COUNTER.

.(15:1)- RESERVEO HOT CURREHTLY USEO.

\*\*NOTE\*\* IN THE RBOVE, SINGLE BIT FIELDS ARE AS DEFINED WHEN THE BIT IS A LOGIC "1".

G.00.00 13-50

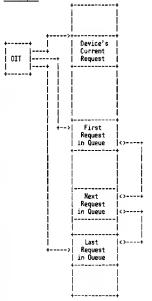
I / 0

#### Disc Request Table and Disc Requests

Requests for disc transfers are effected by acquiring an entry from the Disc Request Table (OISCREOTAB), filling the proper information, and calling the OISCOMANRGER to link the request into the device's doubly linked request qu

queue. The head and tail of a device's request queue are contained in the devices' DIT.

DISCREQUAB



I / 0

#### Disc Request Table

DISCREQTAB DST ENTRY# = 56 (X70) DISCREDTAB PRT = X1017

#### Disc Request Table Entry D Format

DISCREQTABOO	D 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 	
DISCREQTRB01	ENTRY SIZE (X21)	
DISCREOTABO2	PRIMARY ENTRIES	
DISCREQTABO3	IMPEDEO PROCESS PCB	
DISCREQTABO4	TABLE INCEX OF HEAD OF AVAILABLE ENTRY LIST	
DISCREQTABOS	TABLE INDEX OF TAIL OF AVAILABLE ENTRY LIST	
DISCREQTABO6	MAX ENTRIES IN USE	
OISCREQTABO7	CURREHT ENTRIES IN USE	
DISCREQTABO8	OVERFLONS	
DISCREQTRB09	TOTAL REQUESTS	
OISCREQTR810		
OISCREOTAB11	SYSBASE INDEX OF NEAD OF DISABLED REO O	DISCQHEAD
DISCREOTAB12	SYSBASE INDEX OF TAIL OF DISABLED REO Q	DISCOTRIL
DISCREOTAB13	SERIAL WRITE OUEUE HEAD	SERHQNEAD
DISCREOTAB14	A  ////////////////////MAX. SERIAL WRITE OUEUE	A = Active
DISCREOTAB15	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
OISCREQTAB16	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	

#### Disc Request Element Format

Word		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 14 16 15 15 15 15 15 15 15 15 15 15 15 15 15	
Hord	01	REQUEST URGENCY CLASS	URGCLASS
Word	02	LOGICRL DEVICE NUMBER	LDEVN
		MISCELLANEOUS	MISC
		S  DST (IF PROCESS DISC I/O)	
		BRNK (IF SEGMENT TRANSFER)	S=STRCX
Hord	05	OFFSET INTO DATE SEG (IF PROCESS DISC I/D)	RDOR
		RDDRESS IN BANK (IF SEGMENT TRRNSFER)	
Nord	06	UNIT #   FUNCTION	FUNC
No rd	07		XFERCNT
Nord	08		PAR1
Hord	09	P2 (LODR IF SEGMENT TRANSFER	PRR2
Hord	10	////////////////////// OURLIFIER   STRTUS	STRT
Hord	11	FREE  PCB HUMBER	PCBN
liord			PREVREOP
Hord	13	INDEX OF HEXT REDUEST IN QUEUE	NEXTREOP
Ha <b>rd</b>	14	- SEGIOENTIFIER (IF SEG TRANSFER) -	SEGTOENT
Hord	i		
Hord	16	DISPLACEMENT OF RERD OR WRITE FROM SEG BRSE(MM)	SEGDISP

Note: Upon return to free liet, word (N1) becomee  ${\tt Index}$  of  ${\tt next}$   ${\tt EE}$  free entry.

G.00.00 13- 53

Word 0 - QFLRG - Request dependent flags
Bit 0 .ABORT Request has been aborted externally. Bit 1 . MIREQ Request is for a eegment transfer. Bit 2 .DIRG Oragnostic request (not used). System Buffer. Target is a eystem buffer whose index is relative to the etart of the SBUF table. . SBUF Bit 3 . IOHRKE Wake caller on completion of requeet. Blocked I/O. Caller ie waited in RTTACNIO until Bit 5 .BLOCKED equeet ie completed. .COMPLETED Requeet hae been completed and caller woken if he had epecified. Data eegment hae been made precent and is frozen. Bit 7 . DATRERZH . MAMERRORD Bit 8 MAM error on data eegment make present. Bit 9 PREDQUEUED Requeet ie gueued into diec'e reg gueue Bit 10 .SFRIL Start SIO failure in GIP. Bit 11 .PFRIL The I/O has been aborted because of a poperfail. Bit 12 . CURRED Requeet ie device'e current request. Bit 13 . DISABLED Requeet se dieabled. Bit 14 . LDR Request in local ORG. Bit 15 . INLOCAL Buffer DST ie in procese locality. Word 2 - GLDEV.DLDEVN - Logical Device Number Word 3 - QMISC - Device dependent.

Word 4
QOSTM - If SYSBUFRs ie clear then thie ie the DST number of the target data eegment. If bit D ie eet then buffer addreee ie a DB offeet value inetead of segment relative offeet (implemented for MONARIT ID and MOBUFf).

Word 5 QROOR - Offeet in data eegment or sys buff table to target data buffer. Mord 6 DFUNC.FUNC - Function code and qualifiers as specified by driver.

G.00.00 13- 54

I/0

Mord 7

OFFERCHT-On initiation epecifiee the word count if poeitive or byte count if negative. Rt completion of the requeet thie location contains the actual transmission count in the same units as the call. Certain control requeste return data through this

location.

Word 8

SPRR1 - Parameter one, defined by driver
idend 9

SPRR2 - Parameter two, defined by driver
GMISC - Riecellaneoue requeet dependent etorage available to driver.

Nord 10

QSTRT.PCBN - PCB Number of procees which nade this request. Zero if not associated with any process and IOD is to be returned by the eyeten.

.QURLIFIER - R code which further defise or qualifies the general etatus. Defined by driver.

.STRTUS - General Status. Indicates current and result etate of the request according to the following codes.

O - not etarted or awaiting completion.

1 - successful completion.

2 - end of file detected.

3 - unusual condition.

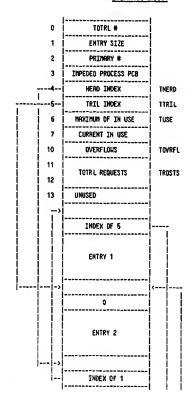
4 - irrecoverable error.

NOTE: See I/O System Status Returne.

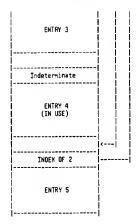
Word 11 - bit 0=1 Q element ie on free liet.

1/0

# IOQ Table Layout



#### IOQ (Cont.)



G.00.00 13- 57

### I/O Queue Element (IOQ)

!	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 	
0	REGUEST DEFENDENT LENGS	QF LAG
1	IOQ POINTER	<b>G</b> LINK
2	QLDEVN	<b>GTDEA</b>
3	MISCELLANEOUS	QMISC
4	S   DATA SEGMENT OST NUMBER	QDSTN S(Word 4(0:1) Stackflag If set QRDDR is DB rel.
5	ADDRESS	QRDDA
6	UNIT   FUNCTION	QFUNC
7	COUNT/XLOG/CONTAOL RETURNS	QWBCT
8	P1	QPRR1
9	P2	QPRR2
10	/////////////////// QUALIFIEA   STATUS	QSTAT
11	FR; PCBN	QPCBN
		ı

QFLAG - Request dependent flags Bit 0 .ABOAT Request has been aborted externally. Special handling is to be applied to this request. For disc, indicates a memory management request. .SPECIAL B1t 1 Bit 2 .OIRG Diagnostic request (not used). System Buffer. Target is a system buffer whose index is relative to the start of the SBUF table. Bit 3 .SBUF Bit 4 . IOWAKE Wake caller on completion of request. Bit 5 .8LOCKED Blocked I/O. Caller is waited in ATTACHIO until request is completed. Request has been completed and caller woken if he had specified. Bit 6 .COMPLETED

G.∞.∞ 13- 58

I / D

I/O Queue Element (Cont.)

Data segment has been made present and is .DATAFAZN Bit 7 .MAMERAORD MAN error on data segment make present. .PREQ This request has been started but was preempted by a NAM request. Bit 9 Bit 10 .SFAIL Start SIO failure in GIP. The I/D has been aborted because of a powerfail. Bit 11 .PFAIL Preemptive type code: 1-soft, 2-hard. Bits12-13 .PREEMPT Bit 15 .NSCODNE A nessage request reply has completed.
QLINK - Table relative index of next IOQ element. Points to first
word of element.
QLDEV - Logical Device Number
QMISC - Device dependent. QDSTN - If SYSBUFRs is clear then this is the DST number of the target data segment. If bit O is set then buffer address is a DB offset value instead of segment relative offset (implemented for NOBUFF).

QRODR - Offset in data segment or sys buff table to target data buffer. GTUNC.FUNC. Function code and qualifiers as specified by driver.

QUBCT - On initiation specifies the word count if positive or byte count if negative. At completion of the request this location contains the actual transmission count in the same units as the call. Certain control requests return data through this location. 

I / 0

#### I/O System Status Returns

170 System Status Hetutis	
Q - PENDING	RTUS X
1 - MAITING FOR COMPLETION 2 - ODING ERROR RECOVERY 3 - NOT RERBY WRIT 4 - NO WRITE RING WRIT 5 - NEW PRPER TRPE WRIT	10 20 30 40 50
1 - SUCCESSFUL	
O - NDRINEL 1 - READ TERMINATED WITH SPECIAL CHARACTER 2 - TAPE RETRY FOR SUCCESS REQUIREO 3 - LOW TAPE OR END OF TAPE RFTER WAITE	1 11 21 31
2 - END OF FILE	
1 - PHYSICAL END OF FILE 2 - DATA 3 - END OF OATA 4 - HELLO 5 - BYE 6 - JOB 7 - END OF JOB	12 22 32 42 52 62 72
3 - UNUSURL CONDITION	
1 - TERMINAL PARITY ERROR 2 - TERMINAL READ ITHED OUT 3 - I/O ABDRETED EXTERNALLY 4 - DATA LOST 5 - DATA SET NOT READY OR DISCONNECT OR UNIT NOT ON LINE 6 - RBOATED BECRUSE OF POMER FAIL 7 - BOOT AND BOS, BSF REQUEST 10 - TAPE RUNAWAY 11 - ECOT AND WAITE REQUEST 12 - NO WAITE RING AFTER REQUEST TO OPERATOR 13 - END OF TAPE (PAPER TAPE LOW) 14 - PLOTTER LITHIT SMITCH RECANED 15 - EMBRLE SUBSYSTEM BREAK AND NO CONTAOL Y PIN 16 - READ TIME RETURNED OVERFLOW 17 - BREAK STOPPED READ 20 - WAITE RIND NO CARD IN WAIT STATION 21 - DEVICE POWERED ON - OPERATING ENVIRONMENT LD	163 173 203
27 - VFC HAS BEEN RESET	273

#### I/O System Status Returns (Cont.)

O - TAVALTO REDUEST

#### 4 - IRRECOVERABLE ERROR

A - THANKTO VEDUESI	4
1 - TRANSMISSION ERROR	14
2 - I/D TIME OUT	24
3 - TIMING ERROR	34
4 - SIO FAILURE	44
5 - UNIT FRILURE	54
6 - INVALIO OISC ADORESS	64
7 - TAPE PARITY ERROR	74
11 - PAPER TAPE TAPE ERROR	114
12 - SYSTEM ERROR	124
13 - INVALIO SOUF INDEX	134
14 - CHANNEL FRILURE, TIMEOUT OR NO RESPONSE FROM	144
CONTROLLER	177
15 - UNINITIALIZED MEDIA (LINUS)	154
16 - NO SPARE BLOCKS RVATLABLE	164
17 - DELETED RECORD DETECTED ON IBM FLOPPY DISC	174
20 - LABELED DEVICE UNAVAILABLE AFTER REELSHITCH	204
21 - PARITY ERROR DETECTED ON PNI COMMANO (EPOC)	214

#### 5 - ERROR IN DATA CONTROL INFORMATION

-		Trate: No. 2 2222	_	HLOU
		INVALIO ITEM NUMBER	5	
		INVALIO ACCESS FOR ITEM	15	VALIO RCCESS
2	-	FAILURE IN FOPEN OR FREAD	25	FS ERROR NUMBER
3	-	PARITY CHANGE IN 8 BIT MODE	35	to amon nonden
4	-	INVALID INFO. FILE FORMAT	45	
5	-	CHECKSUM ERROR IN INFO FILE	55	
6	-	PASSED VALUE LESS THAN MIN.	65	MIN. VALUE BLLOWED
		PRSSED VALUE GREATER THRN MAX.	75	MAX. VALUE RLLOWED
		PRSSED VALUE IS UNSUPPORTED	105	
11	-	COUNT LESS THAN REQUIRED TO	115	MIN.SPACE NEEDED
		RETURN ALL INFQ.		
12	-	COUNT GREATER THAN AVAILABLE	125	MRX. SPACE RVATIL
		TO STORE INFO.		
13	-	PASSED VALUES NOT IN ASCENDING	135	OFFSET OF ELEMENT
		QRDER		
14	-	PASSED CHARACTER HAS DINER	145	OTHER FUNCTION
		DEFINED FUNCTION		

XLDG

G.00.00 13- 61

#### I/O Queue Element for 7976A Magnetic Tape

	0 1 2 3 4 5 6 7	8 9 10 11 12	13 14 15	MNEMONIC
0	Request dependent	flags (see bel	.он)	QFLAG
	SYSDB relative pointer Points to first word o	QLINK		
2	logical devi		DLDEV	
3	RI BI FI GIBOI TOUTI F	SCHTR   BSCHTR	RTCNTR	QMISC
4	S If QFLAG.(3:1) is c DST number of the t S is set, QADDR is i	QDSTN		
5	Offset in the data seg table to the target da	QROOR		
6		Function code this request. next section.	(See	QFUNC
7	On initiation, specific or byte count (<0). At request this location of transmission count in t or words) as in the rec	the tual	QUBCT	
<b>210</b>	Parameter 1. Used only the EOF specification i	for reads. ( n bits (13:3).	ontains	QPRR1
<b>X1</b> 1	Parameter 2. Used only (13:1) is set, writing	for writes. past EOT is al	If bit   lowed.	QPAR2
X12	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	QUALIFIER	STATUS	QSTAT
<b>%</b> 13	PCB NUT	BER	† 	

### QFLAG - Request dependent flags

Bit O ABORT - Rhort this request and return an error indication to the

Bit 1 SPECIAL - Apply special handling to this request. (Not used) Bit 2 DIRG - This is a request from the diagnostic subsystem. bit 2 OTMG - This is a request from the diagnostic subsystem.

(Not used)

Bit 3 SYSBUFF - Target is an index relative to the SBUF Table of

the data buffer.
Bit 4 IDURKE - Wake caller on completion of request.
Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO

Request dependent flags (see below)

IOU table index to the next IOU element.
Points to first word of element.

| | If QFLRG.(3:1) is clear then this is the | S| DST number of the target data segment. If | S is set, QROOR is OB relative.

On initiation, specifies the word count (X) or byte count (Q). At completion of the request this location contains the actual transmission count in the same units (bytes or words) as in the request.

5| Offset in the data segment or system buffer | table to the target data buffer.

I/O Queue Element (IOQ) for CIPER

Logical device number

| Function code for | this request. (See | next section.)

| QUALIFIER |RSTATUS |

PCBN I

QFLAG QLINK

QLDEV

DMISC

COSTN

QADOR

DELING

CUBCT

DPAR1

02882

OPCO

I / 0

I / D

until the request is completed. Implies IOWAKE.

Bit 6 COMPLETED The request has been completed and the caller awakened if he had requested (with TOWAKE).

Bit 7 DATAFRZN Set by the nemory nanagement routines (TMRM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MAMERRORO Set nervo has courred while IMRM was trying to nake the target data segment present and freeze it in nemory.

Bit 9 PREQ Set Not used)

Bit 10 SFRIL Delayed failure of SIO instruction. If a call to STRMT/NPID resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

for execution.

The request was aborted because of a system power failure. Bit 11 PFAIL

CMISC - Driver request dependent flags and counters. Used mostly for

RETRY
BHCK
BHCK
- Backspace record processing for an error retry is in progress.
FORMARD
- Forward space record processing for an error retry is

in progress.

Gap processing for an error retry is in progress.

Backspace record due to a data EOF processing is in BODEDE

TOUTCHTR - GIC timed-out counter.

FSCHTR - Forward space record counter.

BSCHTR - Backspace record counter.

RTCHTR - Error retry counter.

### DSTRT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which nade this request. If zero, the request is not associated with any process and the 100 element is to be returned by the system when the request has completed.

SIRTUS - General status indicating the final state of the request. The following codes are used:

0 - Not started or awaiting completion.

1 - Successful completion.

2 - End-of-file detected.

3 - Unusual, but recoverable, condition detected.

4 - Irrecoverable error has occurred.

CUPLIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

712

%10| Parameter 1.

X11| Paraneter 2.

QFLAG -	Request	dependent flags
Bit 0	RBORT	<ul> <li>Abort this request and return an error indication to the caller.</li> </ul>
B1 t 1	SPECIA	<ul> <li>Apply special handling to this request. (Not used)</li> </ul>
Bit 2	DIAG	- This is a request from the diagnostic subsystem.
	SYSBUF	<ul> <li>Target is an index relative to the SBUF Table of the data buffer.</li> </ul>
81t 4	IOURKE	- Wake caller on completion of request.
B1t 5	BLOCKE	- Blocked I/O. The caller is waited in ATTACHIO

outil the request is completed. Implies IDWAKE.

The request has been completed and the caller awakened if he had requested (unth IDWAKE).

Bit 7 ORTAFRZN - Set by the memory management routines (MMM) when a

G.00.00 13- 63

6.00.00 13- 64

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 MMENDNIC

21

3

MAKEPRESENT request is successfully completed and

0 - Pending

indicates the data segment is frozen in menory.	3 - Not Ready Wait \$30
Bit 8 MAMERRORO - An error has occurred while MAM was trying to nake the target data segment present and freeze	1 - Successful 0 - No Errors 21
it in memory. Bit 9 PREQ — (Not used)	2 - End of File (Not Used)
Bit 10 SFRIL - Oelayed Failure of SIO instruction. If a call to STRRTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected	3 - Unusual Condition 3 - Request Aborted 233 6 - Powerfail Abort 263 221 - Device Powered Up 2213
for execution.  Bit 11 PFRIL - The request was aborted because of a system power failure.	4 - Irrecoverable Error 0 - Invalid Request X4 1 - Transfer Error X14 2 - I/O Tined Out Before Complete X24 4 - SIO Failure X44 5 - Unit Failure X54
QSTAT - PCB number and request completion status.	X12 - Systen Error X124
PCBN - The Process Control Block (PCB) number of the process which nade this request. If zero, the request is not associated with any process and the 100 element is to be returned by the system when the request has completed.	X14 - Channel Failure X144 X21 - Parity Error X214  2608 Line Printer I/O Queue Elenent (HP-IB Systems)
RSTATUS - General status indicating the final state of the request. The following codes are used:	2500 Line Filliter 1/0 doese Clement (in 10 0/3 fens)
O - Not started or awaiting completion. 1 - Successful completion.	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 MNEHONIC
2 - End-of-file detected.	O  Request dependent flags (see below)   QFLAG
Housual, but recoverable, condition detected.     Irrecoverable error has occurred.	1  SYSDB relative pointer to next IOQ element.   QLINK   Points to first word of element.
QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)	2  Logical device number   QLDEV
HP-IB CIPER Physical Driver Request Codes	3 PP PE MC TQUTCHTR    WAITCODE   QMISC
OPERATION FUNCTION PARAMETERS	4   S  If QFLAG.(3:1) is clear then this is the   QDSTN     DST number of the target data segment. If   S is set, QADDR is DB relative.
READ O None URITE 1 None	5  Offset in the data segnent or system buffer   QAODR
FILE OPEN 2 None	Function code for   QFUNC
FILE CLOSE 3 None	this request. (See next section.)
DEVICE CLDSE 4 None	7 On initiation, specifies the word count (>O) QUBCT
CIPER INIT 184 None	or byte count (<0). At completion of the   request this location contains the actual   transmission count in the same units (bytes
CIPER Driver Return Status Codes	or words) as in the request.
General Status (13:3) Qualifying Status (8:5) Overall (8:8)	X10  Parameter 1. Vertical Format specification.   QPRR1   (See next section for detail.)
G.00.00 13- 65	G.00.00 13- 66

I / 0

Z11  Parameter 2. Space Mode Flags. (See next   section for details.)	QPAR2
X12 ////////////// QUALIFIER   STATUS	QSTAT
X13 PCB NUMBER	QPCBN

#### QFLAG - Request dependent flags

Bit 0	HEGRI	- Abort this request and return an error indication to the caller.
Bit 1	SPECIAL	- Apply special handling to this request. (Not used)
Bit 2	OIRG	- This is a request from the diagnostic subsystem. (Not used)
Bit 3	SYSBUFF	<ul> <li>Target is an index relative to the SBUF Table of the data buffer.</li> </ul>
Bit 4	IONAKE	<ul> <li>Wake caller on completion of request.</li> </ul>
	BLOCKED	- Blocked I/O. The caller is waited in RTTACHIO until the request is completed. Implies IOWAKE.
Bit 6	COMPLETED	<ul> <li>The request has been completed and the caller awakened if he had requested (with IOMRKE).</li> </ul>

I / 0

Bit 7 DATAFRZN - Set by the nemory management routines (MRM) when a MRKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MAMERRORD - Am error has occurred while MRM was trying to make the target data segment present and freeze it in memory.

Bit 9 PREQ - (Not used)

Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFAIL - The request was aborted because of a system power failure.

210 230

1 - Waiting For Completion

## QMISC - Driver request dependent flags and counters.

PRE'TO'POST - Pre to post spacing change flag.
PEJECT - Last operation was a page eject.
MASTERCLR - Master clear done to clear powerfail bit in status.
Master clear needs to be done fron not ready condition.
- Channel time-out retry counter.
- Indicates type of wait:
0 - new request
1 - completion wait
2 - not ready wait

## QSTAT - PCB number and request completion status.

USIN: - PLB number and request conpletion status.

The Process Control Block (PCB) number of the process which nade this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

STATUS - General status indicating the final state of the request. The following codes are used:

0 - Not started or awaiting completion.

1 - Successful completion.

2 - End-of-file detected.

3 - Unusual, but recoverable, condition detected.

4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

#### 2608 Line Printer Request Codes

Operation	Function	Parametere
MAITE	1	P1 - Vertical format Specification 1 - use 1st data char as format apec
		X53 - "+", print and suppress spacing X55 - "-", print and triple space X60 - "O", print and double space X61 - "1", print and top of form
		%200-%277, print and space N-%200 lines %300-%377, print with channel N-%277
		All others, print and single space.
		P2 - Space Mode Flags (15:1) - Prespace flag if set, print then fill buffer if clear, fill buffer then print (14:1) - No page stepower flag if set, single and double space uithout stepower (66 lines/page) if clear, single and double space uith stepower (60 lines/page)
FILE DPEN	2	Page eject if not at top of form
FILE CLOSE	3	Page eject if not at top of form
DEAICE CFORE	4	Page eject if not at top of form
READ STATUS	X17	Read I/D status Count - buffer nust be at least 2 bytes
VFC SET	X100	Load VFC ARM Count - form length in words (0 loads RRM form internal ADM) P1 - 6 for 6 LPI or 3 for 8 LPI any other value defaults to 6 LPI
TRB SET	Z1D1	Sets logical column definition P1 - D to 15, any other value defaults to 19

G.00.00 13- 69

# 2619R & 2631 Line Printer IOQ Element (MP-IB Systems)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEMONIC
D	Request dependent flags (see below)	QFLAG
1	SYSDB relative pointer to next IOQ element. Points to first word of element.	QLINK
2	Logical device number	QLDEV
3	PPIPE[PFITOUTCNTR]   PRITCODE	QMISC
4	S If QFLRG.(3:1) is clear then this is the DST number of the target data segment. If S is set, QRDDR is DB relative.	HT2Dp
5	Offset in the data segment or system buffer   table to the target data buffer.	QRDDA
8	Function code for this request. (See next section.)	QFUNC
7	Dn initiation, specifies the word count (x0) or byte count (x0). At completion of the request this location contains the actual transmission count in the same units (bytes or words) as in the request.	QWBCT
X10	Parameter 1. Vertical Format specification. (See next section for detail.)	QPAR1
<b>X11</b>	Parameter 2. Space Mode Flags. (See next section for details.)	QPAR2
X12	//////////////////// QUALIFIER   STATUS	OSTRT
X13	PCB NUMBER	QPC8N
- 1		

QFLAG - Request dependent flags

- Abort this request and return an error indication to the caller.
Apply special handling to this request. (Not used)
This is a request from the diagnostic subsystem. (Not used)
larget is an index relative to the SBUF Table of the data buffer.
Hake caller on completion of request.
Blocked I/O. The caller is waited in RTTRCHID Bit D ABORT Bit 1 SPECIAL Bit 2 DIRG Bit 3 SYSBUFF Bit 4 IONAKE Bit 5 BLOCKED

G.00.00 13- 70

I / 0

Bit 6 CDMPLETED - The request is completed. Implies IOWRKE.

The request has been completed and the caller awakened if he had requested (with IOWRKE). Bit 7 DATAFAZN - Set by the nemory management routines (MAN) when a MIRKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MRMERRORD - An error has occurred while IRMN was trying to nake the target data segment present and freeze it in memory.

Bit 9 PREQ - (Not used)

Bit 10 SFRIL - Delayed failure of SIO instruction. If a call to SIRRITO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution. for execution.

- The request was aborted because of a system power failure. Bit 11 PFRIL

QMISC - Driver request dependent flags and counters for 2631.

PRE'TO'PDST - Pre to post spacing change flag.
PEJECT - Last operation was a page eject.
TDUTCHTR - Channel time-out retry counter.
POWERFRIL - Power fail flag indicates power fail occurred.
WRITCDDE - Indicates type of wait:
D - new request
1 - completion wait
2 - not ready wait

I / 0

Format for 26198

0 1 2 3 4 |PP|PE|PF|TD|BF| | WAITCODE |

TDUT - Channel timed out flag BUF'FILL - Buffer fill operation in progress

QSTAT - PCB number and request completion status.

PCBM - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IDQ element is to be returned by the system when the request has completed. The following codes are used:

D - Not started or awaiting completion.

1 - Successful completion.

2 - End-of-file detected.

3 - Unusual, but recoverable, condition detected.

4 - Irrecoverable error has occurred.

QURLIFIER - R code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

### 2619 Line Printer Request Codes

Operation	Function	Paraneters
WRITE	1	P1 - Vertical Format Specification 1 - Use 1st data char as format specification.
		X53 - "+", print and suppress spacing X55 - "-", print and triple space X60 - "0", print and double space X61 - "1", print and top of form
		%200-%277, print and space N-%200 lines %300-%312, print with channel N-%277
		%320 - Fill Line Printer Buffer Unly
		All others, print and single space.
		P2 - Space Mode Flags  (15:1) - Prespace flag  if set, print then fill buffer  if clear, fill buffer then print  (14:1) - No page stepover flag  if set, single and double space  without stepover (66 lines/page)  if clear, single and double space  with stepover (60 lines/page)
FILE DPEN	2	Page eject if not at top of form
FILE CLOSE	3	Page eject if not at top of form
DEAICE CFORE	E 4	Page eject if not at top of form
RERO STRTUS	<b>%17</b>	Read I/O status Count - buffer size
*IOENTIFY	<b>X110</b>	Return IO value in Bank & Buffaddr
*SELF TEST: INITIATE	<b>21</b> 11	Subtest number to execute in Bank and Buffaddr (subtest number ranges from 0 to 7)
STATUS	2112	Subtest result returned in Bank & Buffaddr
*LOOPBACK TE WRT DATA READ DATA	Z113	Data to LP in Bank & Buffaddr [PING] Data from LP read into Bank & Buffaddr [PONG] Count - Buffer Size (256 bytes nax)

G.00.00 13- 73

### 2631 Line Printer Request Codes (HP-IB)

Operation I	Function	Parameters
WRITE	1	P1 - Vertical Format Specification 1 - Use 1st data char as format specification.
		753 - "+", print and suppress spacing 755 - "-", print and triple space 760 - "0", print and double space 761 - "1", print and top of form
		%200-%277, print and space N-%200 lines %300-%307, print with channel N-%277
		%320 - Fill Line Printer Buffer Only
		All others, print and single space.
		P2 - Space Mode Flags (15:1) - Prespace flag if set, print then fill buffer if clear, fill buffer then print (14:1) - No page stepover flag if set, single and double space without stepover (66 lines/page) if clear, single and double space with stepover (60 lines/page)
FILE OPEN	2	Page eject if not at top of form
FILE CLOSE	3	Page eject if not at top of form
DEAICE CTORE	4	Page eject if not at top of form
AEAD STATUS	<b>X17</b>	Read I/O status Count - 1 byte minimum required
VFC SET	2100	LOROS VFC ARM P1 - 1 - 1 LPI (lines per inch) 2 - 2 LPI 3 - 3 LPI 4 - 4 LPI 5 - 5 LPI 6 - 6 LPI 8 - 8 LPI 12 - 12 LPI Rny other value defaults to 6 LPI.

G.00.00 13- 74

# I / 0

#### I/O Queue Element For HP-IB Card Reader

	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	MNEMONIC
0	Request dependent		OFLAG
1	SYSOB relative pointer Points to first word of	to next IOQ element.	OLINK
2	Logical devi	e nunber	OFDEA
3	Ruxiliary buffer	flag.	ONISC
4	S  If QFLAG.(3:1) is c:   QST number of the to   S is set, QRQQR is (	OOSTN	
5	Offset in the data segr table to the target da		OROOR
6		Function code for this request. (See next section.)	OFUNC
7	On initiation, specific or byte count (0). A request this location of transmission count in or words) as in the rec	QUBCT	
<b>Z</b> 10	Paraneter 1. Contains	the EOF specification	QPRR1
<b>X</b> 11		11:2). (See below card	QPAR2
<b>Z</b> 12	111111111111111111111111111111111111111	QURLIFIER   STRTUS	QSTRT
<b>Z</b> 13	PCB NO	JMBER	QPCBN

#### OFLPG - Request dependent flags

Bit O ABORT - Abort this request and return an error indication Bit 0 ABORT - Abort this request and return an error indication to the caller.

Bit 1 SPECIAL - Apply special handling to this request. (Not used)

1 This is a request from the diagnostic subsystem.

1 Taget is an index relative to the SBUF Table of the data buffer.

Bit 4 IOURKE - Blocked 1/0. The caller is waited in ATTACHIO until the request is completed. Implies IOURKE.

G.00.00 13- 75

Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (with IOWAKE).

Bit 7 ORTAFRZN - Set by the nemory nanagenent routines (TRMI) when a MRKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MRMERRORO - An error has occurred while INRN was trying to make the target data segment present and freeze it in memory.

Bit 9 PAEO - (Not used)

Bit 10 SPAIL - Delayed failure of SIO instruction. If a call to STARNIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFAIL - The request was aborted because of a system power failure.

OMISC - Auxiliary buffer flag used to indicated a read into the driver's buffer and not the user's buffer.

#### QSTRT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which nade this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

STRTUS - General status indicating the final state of the request. The following codes are used:

0 - Not started or awaiting completion.

1 - Successful completion.

2 - End-of-file detected.

3 - Unusual, but recoverable, condition detected.

4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

#### CS 80 Disc Request Queue Element (IOQ)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEHONIC
0	Raquest dependent flage (see below)	QF LAG
1	Request urgency class	OUAGELASS
z	Logical device number	QLDEV
3	CNAMF AS OP IN SA ATRAN LF SP    WAITCODE	QMISC
4	S  OST (If procees disc I/O)	QOSCTN
	DST (If segment transfer) [S=Stack]	
5	Offeet in tha data seg (If process disc I/D)	DADOR
	Address in Bank (If segment transfer)	
6	Unit #   Function code for   thie request.	QFUNC
7	On initiation, epecifiee the word count (%) or byte count (<0). At complation of tha requeet this location contains the actual tranenieeion count in the same unita (bytea or worde) ae in the requeet.	QUBCT
<b>Z10</b>	P1 - Paraneter 1 (Veually Nigh Order of Current Logical Oisc Addreee [CLDA1])	QPAR1
211	P2 - Paranater 2 (Usually Lou Order of Current Logical Qiac Addreee [CLDA2])	QPAR2
X12	///////////////////// QUALIFIER   STATUS	QSTAT
<b>213</b>	PCB	
214	Syebaea relative indx of previous req in queue	QPREVREQP
<b>%15</b>	Syebaaa relative indx of next req in queua	ONEXTREQP
<b>%16</b>	Segidentifiar (If eegnent transfer	QSEGIOENT
X17	Displacement of read or wrt from eeg base (MM)	QSEGDISP
	S   ///////////////////////////////////	

QFLAG - Request dependent flags

GFLAG - Request dependent flags

Bit 0 ABORT - Request has been aborted externally.

Bit 1 MMREQ - Requast is for a segment transfer.

Bit 2 OIGG - This is a request from the diagnostic subsystem.

Bit 3 SBUF - Target is an index relative to the SBUF Table of the data buffer.

Bit 4 IOWAKE - Wake caller on completion of request.

Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTHCHIO until the request is completed. Implies IOWAKE.

Bit 6 COMPLETEO - The request has been completed and the caller awakened if he had requested (with IOWAKE).

Bit 7 OATHAFAZM - Data sagment has been present and is frozen.

Bit 8 MRERRORD - Request is queued into disc'a request queue in nemory.

Bit 9 PREQUEUED - Request is queued into disc'a request queue — Delayed failure of SIO instruction. If a call to STARIIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFRIL - The request was aborted because of a systam power failure.

Bit 12 CURREO - Request is device'e current requeet.

Bit 13 DISSBEED - Request is device'e current requeet.

Bit 14 OISTAIPT - Request is device'e current requeet.

Bit 15 NSGOONE - Reagest request reply hae completed.

QLDEV.QLDEVN - Logical Devica Numbar

QMISC - Orivar requast dependent flaga and counters.

CNAN'ERR'FLD - Channel error retry flag.
RSTRT'FAIL'FLG - Requeet statue failed flag.
OPER'REQ'FLG - Operator requested release flag.
IN'FABLIT'FLG - Intermal maintenance fault flag.
STRT'RTRY'FLG - Statue error eingle retry flag.
RTARMS'FLG - Retransnit required flag.
OPERT - Media load flag.
SYS'PFRIL'FLG - Syeten powarfail flag. WATTCODE

- Indicatee type of wait: O - new request
1 - conplation uait
2 - not ready wait
3 - release/relaaaa dany wait
4 - 100 defer wait
5 - OSCT read wait
7 - eynchronization wait

G.00.00 13- 78

I / 0

QOSTN - If eyeten buffer ie clear than thie ia the QST number of the target data eegnent. If bit 0 ie aet then buffer addraae ie a OB offaat value inetead of eegnent relative offeet (inplanentad for NOWAIT I/O and NOBUFF).

QROQR - Offaat in data eegment or eyeten buffer table to target data buffer.

GFUNC - Function code and qualifiers as specified by

QSTAT - PCB number and request complation etatua.

The Process Control Block (PCB) number of the process which nade this requect. If zero, the request is not secociated with any process and the IOQ element is to be returned by the system when the request has completed.

STATUS - General etatue indicating the final etate of the requaat.

0 - Not started or auaiting completion. 1 - Successful completion. 2 - End-of-file detected. 3 - Unusual, but recoverable, condition detected. 4 - Irrecoverable error has occurred.

QUALIFIER - A coda which further definaa or qualifies the general statue. (See the eection Oriver Return Status Codes.)

I / 0

#### CS 80 Integrated Cartridge Tape Request

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEHONIC
0	Raquest dependent flaga (aee below)	OFLAG
1	Requeet urgency clade	OURGELASS
2	,	OLDEV
3	CHANFIRSIOPIINI RETRY   LF SP    WAITCODE	OMISC
	S! OST (If procese diac I/O)	ODSCTN
	OST (If eegment tranefer) [S=Stack]	
5	Offeet in the data eeg (If procese diec I/O)	QAOOR
	Addrees in Bank (If segment tranafer)	
6	Unit # Function code for this request.	QFUNC
	On initiation, spacifiee the word count (>0) or byte count (<0). At completion of the request this location containe the actual transmiseion count in the same units (bytee or words) ae in the request.	QUBCT
210	P1 - Parameter 1 (Usually Nigh Order of Current Logical Diac Addresa [CLDR1])	OP AR1
<b>211</b>	P2 - Paraneter 2 (Usually Lou Order of Current Logical Diec Addresa [CLDR2])	OPAR2
<b>%12</b>	PCBN QUALIFIER   STATUS	QSTAT
<b>%13</b>	Sysbase relative indx of previous req in queue	QPREVAEOP
<b>X14</b>	Sysbase relative indx of next req in queue	QNEXTREQP
<b>%</b> 15	Segidentifier (If eegment transfer	OSEGIDENT
<b>%</b> 16	Displacement of read or urt from eeg base (MM)	QSEGDISP
	S ////////////////////////////////////	

OFLAG - Request dependent flags

Bit 0 ABORT - Request has been aborted externally.

Bit 1 MIREQ - Request is for a segment transfer.

Bit 2 DIRG - This is a request from the diagnostic subsystem.

Bit 3 SBUF - Target is an index relative to the SBUF Table of the data buffer.

Bit 4 IOUARKE - Hake caller on completion of request.

Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTRCHIO until the request is completed. Implies IOUARKE.

Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (uith IOUARKE).

Bit 7 DATIRFRZN - Data segment has been present and is frozen.

Bit 8 MRDERRORD - Rerror has occurred while IRMI was trying to make the target data segment present and freeze it in nenory.

Bit 9 PREQUEUED - Request is queued into disc's request queue and to SIGNII - Delayed Failure of SIO instruction. If a call to SIGNII resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFRIL - The request was aborted because of a system power failure.

Bit 12 CURREQ - Request is device's current request.

Bit 13 DISABLED - Request is disabled.

Bit 14 OISAITMPI - Retrept to disable this request.

Bit 15 RSGOME - R message request reply has completed.

QUDEVN - Logical Device Number

QLDEV.QLDEVN - Logical Device Number

 ${\tt QMISC}$  -  ${\tt Driver}$  request dependent flags and counters.

CHRN'ERR'FLG - Channel error retry flag.
RSTRI'FAIL'FLG - Request status failed flag.
OPER'REQ'FLG - Operator requested release flag.
IN'FAULT'FLG - Retry count area.
LNBO'FLG - Metal load flag.
SYS'PFRIL'FLG - System powerfail flag. WAITCODE

- Indicates type of wait:

0 - new request
1 - completion wait
2 - not ready wait
3 - release/release
4 - IOO defer wait
5 - DSCI read wait
6 - DSCI write wait
7 - synchronization wait

QDSTN - If system buffer is clear then this is the DST number of the target data segnent. If bit O is set then buffer address is a DB offset value

instead of segment relative offset (implemented for NOWRIT I/O and NOBUFF).

QRODR - Offset im data segment or system buffer table to target data buffer.

QFUNC - Function code and qualifiers as specified by driver.

QSTAT - PCB number and request completion status.

 The Process Control Block (PCB) number of the process which wade this request. If zero, the request is not associated with any process and the TOQ element is to be returned by the system when the request has completed. PCBN

STATUS - General status indicating the final state of the request.

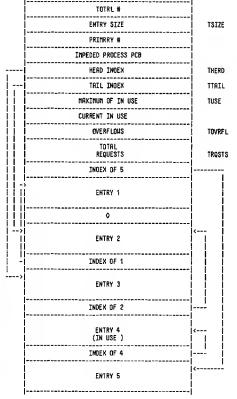
O - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.

QURLIFIER - A code which further defines or qualifies the general status. (See the section Oriver Return Status Codes.)

G.00.00 13- 82

I / 0

SBUF Table Layout



I / 0

#### Table Element Allocation (SBUF)

The allocation of the elements in the IOQ terminal buffer (TBUF) and system buffer (SBUF) tables is of concern to the I/O system.

FREE LIST OF TABLE ELEMENTS

These tables are in the form of a free-linked list of the free elements. For the SBUF's the -1 word of entry is the link to the next element. For the TBUF's, word zero is the link and word 1 is the link for the IOQ elements.

Each word has an 11-word header beginning at the base of the table . The first six words of the header are for managing the table and the second five are for monitoring table activity.

The entries follow the header at word eleven.

#### **ELEMENT RLLOCATION**

Elements are obtained from the beginning of the free list, pointed to by the head and returned to the end of the free list pointed by the tail.

When the free list is empty, the head index is zero and the tail index is set to point at the head index.

The tables are divided into two areas: a primary and a secondary area. Most requests are obtained from the primary area. The secondary area is used only for critical requirements when the primary area is exhausted. These areas are logical areas determined by parameters in the header.

The utility of the core resident tables is seriously reduced if their use is not restricted to dynamic situations.

One of three responses must be specified to the routines which allocate elements from the  ${\rm I}/{\rm O}$  system tables.

- 1. Inpede caller if primary is empty.
- 2. Get from primary area only.
- 3. Get from secondary area if primary area is empty.

3 - 1 - 5 - 4 - 2

G.00.00 13-83

#### Table Element Allocation (Cont.)

Request types 2 and 3 return an indication to the caller if the request could not be satisfied. The following table specifies the types of calls for element allocation and the action if an element is not activated.

BUFFER USER

CRLL TYPE

FINAL ACTION

file system Ptape Bad track

Impede Impede Primary

forget request

RTTACNIO (not impedable) RTTRCHID (impedable) SIODM (memory management) IDMESSAGE

Primary Impede Secondary Secondary

Return IOQX-0 Sudden death I/O error

NERDER DEFINITION

27

Primary # - Mumber of elements in the primary area.

Total # - Total number of elements in the table.

Size in words of each element.

- If not zero then contains the PCB number of the first process waiting for an element in this table.

Head index - Index of first free element.

In use - Current number not in free list.

Number of requests nade for an element.

Total number of elements requested.

G.∞0.∞ 13- 85

ICS Global

63. RESERVEO 50 49 CANOPIN 48 LAST WEIGHT 47 PRUSETIME LISTSTRIE 45 CUREFILTER 44 **CUADFILTER** 42 CUTNUM

CHTDENON CURCFILTER

MAXCFILTER 39 MINCFILTER 38

37 ESCHEOBASE DSCHEDBASE 36

CSCHEDBASE MORSTEPRI WORSTOPRI 33

32 WORSTOPRI MISC. BOUNDS FLAGS

SYSTEM MEM BOUND 29 XOS UPPER BOUND 28 DL INITIAL

6.00.00 13- 86

I/D

I / 0

08 BRNK RETURN FOR DISPRTON DB AETURN PARM

P=PSEUDO-DISABLED AND DISP INSTRUCTION EXECUTEO. D=DTSPATCHER INTERRUPTED.

26 XOS SEGMENT BANK Series 64 only 25 **XOS SEGMENT BASE** Series 64 only 24 XDS SEGMENT LIMIT Series 64 only 23 PRIV BNDS STRT UD Series 64 only 22 RESERVED 19 DISAP PSEN, PSOB counter Reserved SDST process' stack DST# 15 PSTA pseudo-interrupt status PADDR pseudo-interrupt address TAACE FLAG flag set non-zero on IXIT away from ICS PERTI. PTR to powerfail PCB JCUT absolute JCUT address ΧP pointer to executing process PCB PCBX absolute stack address stack DB relative Z DL stack DB relative DL s stack DB relative S SBANK stack bank

G.00.00 13- 87

absolute stack DB

DISPRICH stack marker

STDB

0

STATUS

0

O P

G.00.00 13-88

#### ICS Global Cells With Initial Values

STDB - absolute address of the currently running process's stack.

SBRNK - bank address for process' stack.

- stack DB relative DL

- stack DB relative DL

- stack DB relative Z

PCBX - absolute stack address

KP - PCB table relative pointer to word 0 of the running process'

PCB.

The above cells are to be initialized for the PROGENITOR.

CPCB - absolute 4, is an absolute version of XP. If CPCB is zero, then the above cells are invalid. This will never be the case in a process. CPCB should also be set by INITIAL.

SDST - DST# for running process' stack.

JCUT - the bank zero absolute address of the JCUT table.

PRDDR - PB relative address for the procedure PSEUDDINT.

PSTM - status value for PSEUDDINT, X140000+CST#.

DISRP - PSDB counter, initially 0.

INITIAL sets the above as described.

G.00.00 13- 89

T / D

The first word is used by the channel program processor to store status information after  $I/\theta$  channel aborts. The next word is used by the driver to indicate if status should be examined for special conditions or errors. The other two words are not used.

ICPVR4 - DMR abort address

If a DMR abort occurs, the absolute address where the abort occurred is stored in this area.  $\,$ 

ICHTRL - Contains controller information

-If this bit is set, the controller is sharing a software channel resource in order to limit bandwidth.

CHANQUE - The software channel resource number.

-Channel number (four most significant bits of DRTH).

DEV -Device number (three least significant bits of DRTH).

IQUEUE - The channel program contains:

SIDPSIZE - (number of words + 1)/2 in the channel program area.  $\ensuremath{\mathsf{CQUEN}}$  - or a multi-unit controller this field contains the software controller resource number.

IFLAG - Controller and Channel Program state flags

RUHWAIT - An Idle Channel Program should be started when there are no active requests to process.

WRITPRDG - An Idle Channel Program has been started for this controller. This bit is reset by an interrupt.

IGMOREHI - An HIDP instruction has been issued against this controller but the channel program was not in a wait statement. Therefore ignore the interrupt generated by

HCUHIT the channel code when this program halts. - Highest configured unit number for this controller.

ISTAT - 20 bytes of status from the idle channel program.

#### CS 80 Disc Interrupt Linkage Table (ILT)

There is one ILT for each device controller configured on the system. A controller nay support more than one unit, however the CS'80 disc driver will only concern itself with the single unit controller.

+-	0 1 2	3	4 5	6	7 8	9	10 11	12	13 14 15	. 1	MNEMDHIC
0  1  2  3		Char	nel Prog	ıran Vari	able	!	ICPVA)			 	ICPVRO ICPVR1 ICPVR2 ICPVR3
4   5			Abort Addr	pee							ICPVA4 ICPVA5
6					0					ĺ	ISRQL
7 L	II (	CHANG	UE	į		Ĭ	CHAN	<b>!</b>	DEV	i	ICNTRL
Z10]	SYSOB n	elati	ve po	inte	r to	ch	annel p	rog	ran area		ISIOP
Z11  3	SYSDB re	elati	ve po	inte	r to	id	le stat	us	area		ISTRP
	the devi	ice u	nit n	unbe	r fr	on '	the cta	tue	extract pointed on the Unit 0]		IUNIT
0	YSDB re urrent] data ope	ly us	ing t	I po	hann	el '	to perf	OTH	ce	•	ICDP
714	\$I	DPSI	ZE		<del></del>		CQUE	N			IQUEUE
715 R	UP IG						í	н	CUNIT		IFLAG
	YSDB re								 		IDITPO
217	0 bytes	sta	tus a	rea	for	idle	chann	 el	progran		ISTAT
· •									 		
. 1									<del>-</del> +		
z31   				Dis nnel gran					i		

ICPVAO - Channel Program Variable Area

#### CHRPTER 14 SPOOLING

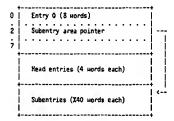
### Input Device Directory/Output Device Directory

IDD/ODD (Common attributes referred to as XDO)

IDD: DST = 45 (= X55) SIR = 3

DDD: DST = 46 (= X56) SIR = 4

### Overview of Table Structure



G.00.00 14-1

Spooling

### Typical Head Entry (4 words)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15					
Device outfence  ////////////////////////////////////					
Head pointer					
Tail pointer					
Logical device					

There are two types of head entry, a class entry and a logical device entry. There is only one class entry, if it exists at all, and it is the first head entry in the XOD. Rll spoolfiles opened by class (e.g., iP, SLOHUP, EPOC. PP, etc.) are linked to this entry. There is one logical device entry for each real (physical, as opposed to virtual) device on the system. Output devices appear in the DDD, input devices in the IDD. RC/DC devices such as terminals appear in both directories.
Each head entry is linked to 0 or nore subentries (a typical subentry is shown in the next table). R null chain (0 subentries) consists of head pointer = 0 and tail pointer = segment-relative address of the associated head pointer. If one or nore subentries exists, the pointere are segment-relative addresses of the first word of the first and last subentries of the chain. Rny intermediate subentries are linked through the subentries. The tail subentry always contains a 0-link.
The Device Dutfence and LDEV# fields are meaningless for the class entry. For logical device entries (non-C Logical Device field), a non-O Device Dutfence neans that this outfence overrides the system-wide outfence in word 4 of entry 0, but only for this device.

#### Entry O (Overall Table Definitions)

Spooling

0 1 2 3 4 5 6 7 +       0  Maximum size		
1 Head entry size = 4	Subentry size = %40	1 ( Hords )
2  Subentry area pointe	r (segment relative)	12
3 DD  Hext avail dev	ice file ID (DFID)	Ĭ3
41/////////////////////////////////////	//////// Fence	<b>1</b> 4
s <i>\(111111111111111111111111111111111111</i>	mmmimmm	įs
6	///////////////////////////////////////	<b>1</b> 6
?\!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	7

0 ==> This is the IDD, 1 ==> This is the ODD.

Fence: For spooled output devices (000), the system-wide out-fence. For spooled input devices (IDO), the jobfence.

G.00.00 14- 2

Spooling

#### Tunical Cubandani (750 usada)

0 1 2 3 4 5 6 7 8 9 1	+
20 // State  Outpri  CL //////	
X1 Type Job number	er <b>[</b> 1
72  73  User name 74  75	2  3  4  5
X6  X7  Account name X10  X11	6  7   <b>8</b>  9
212  213  Job name 214  215	10  11  12  13
X16  X17  File name X20  X21	114 115 116 117
X22 ID  Device file ID	
Z23 FS DR //  XDD head index (see	explanation)  19
X24  Logical device, or Device Cla	ass Table index  20
X25  Virtual LDEV number of open s	spoolfile  21
X26  Volume table index   Sect	
227 of s	spoolfile label.  23
230   Number of extents  //////	/////////////////////24
X31  Last extent size (se	ectors)  25
X32 SQ // RS F0 S0 RB //  Humbe	er of copies   26
X33  Segment-relative link to me this device or class. 0 ==>	ext subentry,  27
X34  Number of records in spoolfi X35	ile (doubleword)   128  29
X36  Year NOO 100   Julian	Day of Year/2 130
237 DY  Hour (24 hr)   Minute	Seconds/4  31

Note: Worde 0-X24 are used in all subentries. Words X25-X37,

although precent in all embentries, are zero unless the subentry is for a spooled file (spoolfile).

3 ==> Locked

CL --- 1 ==> Word X24 is a clase index into the
Device Class Table.
0 ==> Word X24 is the LDEV aeeociated with
thie eubentry.

Word 1: Type -- Describes which environment created the

Describes which environs
eubentry:

O ==> Session' (SPOOK)

1 ==> Session

2 ==> Job

3 ==> Job' (SPOOK)

| See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See | See

SD

device.

- Spaced Dut bit. File System could not acquire a new extent when creating spoofile.

- This is the \$\$TOLIST of an aborted job.

- Time stamp when epoofile was nade READY, or DD if not closed properly. Julian day is 9 bite etarting with Word X36, bit 8. Words 236-37:

G.00.00

Specting

#### SPDOK Tape Format

The overall format of output tapee produced by the SPDOK "DUTPUT" command le ehoun below. The varioue conponente of the tape are then described in detail. The format described here is subject to change as MPE evolves. Riso, there may be errors in SPDOK which would cause the actual tape format to differ from the one described here in eone cases. All numeric information ie in integer format unless otherwise specified.

EOF

EDF

Label Record

File Directory Records

Device and Clase Directory Record

Spoolfile

EDF

Spoolfile

FDF

Mechaniens for End-of-tape and tape switching are the eame ae for STORE/RESTORE tapes.

#### Label Record

Worde 0-13: "SPOOLFILETAPE LABEL-HP3000."

reel number (first reel ie number 1)

date (from CRIENDAR intringic) Mord Words 25826: time (from CLOCK intrineic)

"MPE V" if an MPE V SPOOK tape Marde 30831:

G.00.00

Spooling

All other worde are zero.

#### **file Directory**

The file Directory hae one entry for each epoolfile on the tape. Each entry is 12 words, and entries are packed into as many 1020word records as needed. The last record will be padded with zeros if necessary. The entry format is:

Nord Đ: Device file id number (bit 0 is on to indicate that the file ie an output epoolfile)

Norde 1-3: zero

Norde 4-7: User name

Worde 8-11: Rccount Mane

#### Device and Class Directory

The Device and Claes Directory is contained in one 1D24-word record. There is no EOF separating this record from the File Directory. This directory contains one entry for each logical device or device class linked to the epoolfiles on the tape. Rleo, there is an entry for each logical device in each claes in the directory, whether or not that logical device was directly referenced by a epoolfile. The entries are packed into the tape record one after another in no particular order. The entry formate are shown below.

### Logical Device Entry

Word 0: logical device number

Mord 1:

Bits 0:8: device subtype Bite 8:8: 3 (=length of thie entry in worde)

Nord 2: device type

Speciing

#### Device Clase Entry

O: Device clase number (negated). This ie the number of the entry of thie device clase in the eystem's Device Clase Table.

Vord 1: Total number of worde in this entry.

Worde 2 on: The entire contente of the Device Claes Table entry for thie device class.

#### Spoolfile Format

DDD entry (32-word tape record)

Spoolfile block ---> Two epoolfile blocke packed into one Spoolfile block 1024-word tape record.

Two spoolfile blocks

Two spoolfile blocks

The first few spoolfile blocks have been modified to contain user label information from the epoplfile. This is explained later,

### Spoolfile Block Format

A spoolfile block ie a 512-word block that containe variable length records in spooler format. Spoolfile records start at the first word of the block. The last record ie followed by a -1 to indicate that no more records follow. The last two words of the block contain a doubleword which is the record number of the first record in the block.

### Spoolfile Record Format

Byte count of record - 2

Word ': Byte count of data portion of record. Note that thie count includes trailing blanks. Nowever, trailing blanks are truncated in

G.00.00

G.00.00

#### Spooling

the actual record, so this count may be nore than the number of bytes actually present in the data portion.

Nord 2:

Function Code: 1=Furite 2=Fcontrol 3=Fopen 4=Fclose

##FCLOSE ## X100 and beyond=FDEVICECONTROL

Word 3: P1 -- RTTACNIO parameter

Word 4: P2 -- RTTRCNIO parameter
Words 5 on: Data Portion of Record

#### User Labels Information

- 1. Master: user label 0.
- 2. FOPEN entry catalog: user labels 1-10.
- Circular queue for restart checkpointing: user labels 11-27.

Since older versions of MPE did not use user labels, a way was needed to incorporate them into the SPOOK tape format without losing forward and backward
compatibility. The method used is to add several special spoolfile blocks to
the beginning of the spoolfile on tape. Each of these blocks has exactly one
FOPEN record at its beginning. This record is followed by a -1. Thus old
versions of MPE will assume that the rest of the block is garbage. Nowever,
the rest of the block is actually used to contain user label information.
The first two spoolfile blocks (i.e. the first tape record of the spoolfile
proper) contain only the FOPEN records. The next 5 tape records actually
contain user labels in addition to the FOPEN records. The user labels are
packed 3 to a spoolfile block, by to a tape record. Each spoolfile block of
512 words has the following format:

Words

-4: FOPEN record

Word 5:

-1 (to "terminate" the block)

Words %200-%377:

user label

Nords X400-X577:

user label user label

Mords %600-%777:

c ^^ ^

G.00.00

Spooling

Following this special group of blocks, the spoolfile resumes a normal format. The special FOPEN records all have the number of user labels in P2.

It is often the case that some of the 27 user labels have not been initialized before the tape is written. In that case, their places will be filled with garbage. There is no easy way of detecting this except by careful inspection.

G.00.00 14- 10

#### CHAPTER 15 UNIFIED COMMAND LANGUAGE (UNCL)

# Reply Information Table (RIT) DST X34: SIR X25

٧	1	
ô	NUMBER OF ENTRIES Flag≈10	1
1	MAX MUMBER OF ENTRIES	
5	POSITION OF NEXT FREE ENTRY SPACE IN QUEUE	TABLE 57
3	NUMBER OF QUEUED ENTRIES	I REMUER HE
	(52 HORGS TO HOLD PINH'S OF QUEUED ENTRIES)	
Q	UHUSEO (PIH)	}.
4	OST# (FOR REPLY)	Ì
5	BUFFER RODRESS (OST RELATIVE)	
3	MAX LENGTH OF STRING   REPLY TYPE EXPECTED	
4		İ
5		į
5	  lag=1	ENTRY
7	# SYTES IN MESSAGE	(51 ude)
	Message in Ascii	
.1	14g=1 (UP TO 85 CHARS.)	
		1

HOTE: Procees Number = O means entry is empty
Reply Type = O for number (num)
= 1 for yes or no (y/n)
= 2 for string (sxx)
= 3 for yes, no, or STRING

G.00.00 15- 1

### Unified Connend Lenguege

.flag=2

TABLE SIZE = 2046 words

.flag=2 MRX # OF ACTIVE ENTRIES = 39 MRX # OF QUEUED ENTRIES = 52

#### Message System General Description

- The message system comeists of the following parts:

   Celleble intrinsic DEMMESSAGE.

   Uncelleble procedure DEMMESSAGE.

   System message catalog (CATALOG.PUB.SYS) and any number of user catalogs.

   Program HAKECAT which builde message catalogs.

   MESSAGE SYA X24

   MESSAGE SYA SCALOR CELLS X371-373

   MESSAGE DATE GETTERY

- MESSAGE DATA SEGMENT

The message system is used by calling GEMMESSAGE (or GEMMSS) with a message number. The message system fatches the message from a message catalog, inserts paremeters, then routes the message to a file or returns the message in a buffer to the caller.

A nessege catalog is a numbered editor—type file containing eate of messages. The sets serve to break a catalog into manageable portione. A message system user may cell GEMESSAGE using either his own message catalog or using MPE's catalog (CRTRLOG.PUB.SYS).

After creeting a message file, run the program MAKECRT in order to build a catalog that is readable by the message system. This file is still readable by the aditor (it can be "texted") but it contains a directory (uritten se e userlabel).

In order to use the message catalog, the program must first open the message catalog, then call GEMMESSRGE with the file number, sat number and message number. (RPE users don't need to open the catalog, GEMMESS autonatically uses CATRLOS, PUB.SYS.) The file numb be opened with the apptions "MOBUF" and "MULTI" -record

G.00.00

# Unified Command Language

#### Message Catalog

heseages in the cetalog can be of any length and can contain up to five parameters. Continuation of a message is indicated by """ or "\"" at the end of a line. The """ evoloul indicates that the message is continued and that a carriage return, line feed be issued the terminal. The "\"" symbol indicates that the message is continued on the same line with no carriage return, line feed.

Parameters may be inserted into the message fattern, line feed, catalog. The parameters are passed in the GENMESSREE (or SEMISG) call and inserted wherever a "I" is found. For the system reassege catalog, the back slash () is also a parameter, reflecting a logical device number. The message is routed to the user associated with that logical device through the inSOUIRHE command, Reseage sets are indicated by "SSE" in 'starting in column 1 (the rest of the line is treated as a comment). Haximum value for n is SJ. Commente can be inserted in the catalog by placing "S" in column 1. Reseage numbers are positive integers, need not be contiguous, but must be in ascending order. RF ther processing by the program MRECRI, the catalog file contains records of 30 bytes, blocked 16, in 32 extents. (The system message catalog is only one extent, however). The format of the message catalog is as follows:

\$2EL 4 SAZEU WESCHGES
2 TO MIN OF FORM LENDING
5 TOEAN #1 IH RISE BA DIVENORZICS
1 TOEAN #1 IH RISE BA DIVENORZICS
1 TOEAN #1 IH RISE BA DIVENORZICS
2 TO MIN OF WESCHGES
2 TO MIN OF WESCHGES
3 TO MIN OF WESCHGES
4 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MIN OF WESCHGES
5 TO MI

# NESSAGE 35 IS TWO LINES LONG, R PARAMETER STARTS THE # FIRST LINE AND THE SECOND LINE IS "NP32002"

35 !X HP320028.00. !

276 LDEV # FOR "!" OH ! (HUN)! SET 2 CIERROR NESSAGES 82 STREAM FACILITY NOT ENABLED: SEE OPERATOR. (CIERR 82) 800 NORE THEN 30 PARAMETERS TO BUILD COMMAND. (CIERR 800)

204 FILE COMMAND REQUIRES AT LEAST THO PARAMETERS. INCLUDING

#### Unified Command Language

FORMAL NAME OF THE FILE (CIERA 204)

## MRKECHT Program

The progren MAKECAT.PUB.SYS is used to build message catalogs (and also HELP catalogs). The program's input file has the formaldesignator IMPUI, which nust be used for all entry points. The program has the following entry points:

Reads from input file and builds a temporary file (formaldesignator CATRLOG). Also rememes any old temporary CATRLOG, CRInn, using an archival numbering scheme (i..., CRII, CRIZ, etc.).

BUILD - (Nust log on under MRMRGER.SYS.) Reads from input file, build the system message cetalog (formaldesignator CRTRLOG), and instells the message system. Existing catalog is remamed CRTmmm according to the same scheme as for no entry point (above). Installation of the message system means moving the directory contained in the userlabel of the catalog into a date segment. The DST mumber and the disc address of CRTRLOG ere placed in system global area. The message system may be installed while the system is running.

- (flust have PM or OP capability.) Installe the system message catelog (does not build a new one). Opens arout file, moves the directory in the CRTMIDD into e date segment, and places the OST number and disc address of CRTMIDD in system global area. This may be done when the message system seems to be "broken", but the catalog is intact. (MPC is issuing "MISSIMO MSS. SETwem. MISSIMO" at terminals and at the console.) This may be done while the system is running. OTA

Used to build the HELP catalog. Reads input file and builds a HELP catalog (formaldesignator HELPCRT). RELP

#### Unified Command Language

#### Message System CATRLOG. PUB. SYS

SSET 1 - System messages.

SSET 2 - Ci errors and marnings messages.

SSET 3 - Miscellaneous RBURT messages.

SSET 4 - Program error abort messages.

SSET 5 - Intrinsics abort messages.

SSET 6 - Run-time abort messages.

SSET 7 - CI general messages.

SSET 8 - File System error messages.

SSET 9 - Loader error messages.

SSET 10 - CRERIE error messages.

SSET 11 - MCITWATE error messages.

SSET 12 - SUSPEND error messages.

SSET 13 - MTCOMRND error messages.

SSET 14 - LOUKGURIN error messages.

SSET 15 - Private Volumes error messages.

SSET 16 - Gos/3000 messages.

SSET 17 - HELP Facility error messages.

SSET 18 - Graphic devices messages.

SSET 19 - Serial Disc error messages.

SSET 20 - User Logging error messages.

SSET 21 - Resociation Utility (RSOCIABL) messages.

SSET 22 - C&SOR Page Printer nessages.

SSET 25 - C&SOR Page Printer error file messages.

SSET 25 - C&SOR Page Printer error file messages.

SSET 25 - C&SOR Page Printer nessages.

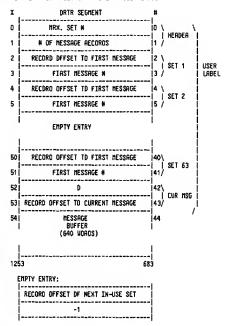
G.00.00 15- 5 Unified Command Language

#### Message Set Directory

OST # IN SYSGLOB %373

CAT DISC ADDA IN SYSGLOB #371-372

CREATED BY AUNNING MAKECRT. PUB. SYS.
KEPT IN A ORTH SEGMENT AND IN A USER LABEL.



G.00.00 15- 6

## Unified Connand Language

### NELP Subsystem

KEPT AS USER LABEL AEAO ONTO USER'S STRCK USES SEARCH INTRINSIC FORMRI VARIRBLE ENTRY SIZE

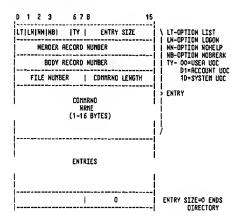
7		
D	OIRECTORY SIZE (WORDS)	
1	ENTRY LGTH (BYTES)   KEYWORD LGTH (BYTES)	<b>y</b>
2	ENTRY Keynord	ENTRY
	1-255 BYTES	
	ENTRY RECORD # IN CICAT LEFT BYTE   RIGHT BYTE	,
	ENTRY LGTN (BYTES)   KEYHORD LGTN (BYTES)	Y
	ENTRY KEYWORO 1-255 BYTES	ENTRY
į	ENTRY AEC # LEFT BYTE	
l	ENTRY REC # R. BYTE   ENTRY LGTN (BYTES)	l y
	KEYHORO LGTN (BYTES)	-
	ENTRY KEYNORO 1-255 BYTES	ENTRY
į	ENTRY REC #	ļ
į	LEFT BYTE   RIGHT BYTE	1
	ı	

-----

## Unified Connand Language

#### UDC Directory

\*EXTAA DATA SECMENT - DST # IN DB+Z255 OF UMRIN STRCK
\*BUILT BY INITUDC



#### Unified Command Language

### UDC's COMMRNO.PUB.SYS

\*RECDRO SIZE = 20(10) MOROS, 6 RECOROS/BLOCK \*KEEPS TRRCK OF WHD IS USING WHAT UOC CATALOG \*CRN BE PURGED TO DISRBLE UCC'S

\*CAN BE REBUILT TO RE-ENRBLE UDC'S

X,	RECDRO O	#	z	FREE ENTRY #	ŧ
٥	1st FREE ENTRY #	0	0	NEXT FREE ENTRY # 0	)
1	not used	1	1	ENTRY TYPE=0 1	
2	MRX IN USE	2	2	 	•
3	# IN USE	  3		not used	
4		4		l į	
- 1	not used	l			
- 1		1			
23		19	23	 	9
23		19	23	i   	1

G.00.00 15- 9

### Unified Command Language

### COMMAND. PUB. SYS (Cont.)

Z	USER ENTRY	#	Z	FILE ENTRY	*
0	CATALOG ENTRY #	0	0	NEXT CAT. ENTRY #	0
1	ENTRY TYPE=1	1	1	ENTRY TYPE = 2	1
2		2	2	     FILE NAME	2
3		3	3		3
4		4	4	FDPEN FORMAT: 	4
5		5	5		ļ  5
6		6	6	FILE	6
7	ACCOUNT*	7	7	(/LOCKHORD)	7
10		8	10	GRDUP	8
11		9	11	ACCDUNT	9
12		10	12	٥	10
13	not used	11	13		11
14		12	14		12
15		13	15	(UP TD 36 BYTES)	13
16		14	16		14
17		15	17		15
20		16	20		16
21		17	21		17
22		18	22		18
23		19	23		19
		ı			l

\* IF THE USER FIELD AND THE RCCDUNT FIELD CONTAIN "9\_\_\_\_THIS INDICATES SYSTEM LEVEL UDC'S.

IF DNLY THE USER FIELD CONTAINS 0 AND 7 SPACES, THIS INDICATES ACCOUNT LEVEL UDC'S.

G.00.00 15- 10

#### Unified Command Language

## CI Stack Definition

0B+X0	BCDMIMAGE (Byte Ptr. To Command)
0B+%1	
08+7215	LINELENSTACK \\ (30 words) \\
00.000	
0B+Z253	NEXTHSG (Not currently used)
08+2254	THIS IS SPRRE
08+2255	UDCO
08+2256	UDC1
08+2257	UDC2
DB+%260	UDC3
DB+%261	UDC4
DB+X262	IFMESTING
DB+#263	IFSKIP
DB+2264	ELSESEEN
D8+Z265	CIFLAGS
DB+%266	CONTINUE STATE STACK (2 words)
08+%270	PENDINGCOMLEN
DB+%271	BLASTCOMIMAGE (Byte Ptr.)
0B+%272	LAST COMMRNO IMRGE (280 bytes)
	<u> \ \</u>

# Unified Command Language

#### Field Definitions

BCDMIMAGE: Byte pointer to CDMIMAGE (sometimes called MCDMIMAGE) in the CI stack.

CDMMAND IMAGE: Command character string currently being

LINELENSTRCK: A CI command can span up to 30 input lines. This stack holds the length of each input line.

NEXTMSG: Used to be used to link messages together. No longer being used.

THIS IS SPARE: Not used.

UDCO: Holds the OST number of the UOC definitions.

UDC1: Holds the old S register value for UDC's.

UDC2: (0:1)--FLUSHUOC, used by :SETCATALOG

UDC3: UDC options for current UOC.

UDC4: (0:1)--UDC Fatal C1 Error (1:1)--UDC EXITBREAK (2:1)--UDC BREAKDETECTED (3:1)--UDC NDPRINT (4:1)--UDC INRCENDIUST (10:6)--UDC WESTLEVEL

IFNESTING: Level of nesting of :IF commands.

IFSKIP: Whether the current commands are being skipped as the false part of a :IF command.

ELSESEEN: Level of the :ELSE commands.

CIFLAGS: (13:1)--Sequenced: line numbers at rear. (15:1)--Hot REODable (last command).

CONTINUE STRTE STRCK: History of the :CONTINUE commands. = 0--no :CONTINUE = 1--just seen = 2--in effect.

BLASTCOMIMRGE: Byte pointer to last command image.

LAST COMMRNO IMAGE: When a command completes execution, the command string is copied here for use by the :REDD command.

# Association DST Layout

	o	DST 242
i	1	
Not I	3	SIR X30
Used	4 5	0
1	5 6	One entry/ system ldev
	•	System IDEY
JHAT Index	7	1
JII DST Number	8	ļ
JII DSI MUMBER	۰	
OST rel, index to user's next entry.	9	- Ldev 1
		(Associated)
Class name under which this ldev is	10	Ì
associated. Left justified and	11	ļ.
padded with blanks. 8 bytes.	12 13	}
	13	,
i o i	14	1
		ĺ
! • !	15	!
0	16	  - Ldev 2
	10	I COEV E
•		(Unassociated
1	17	}
Don't	18	
Care	19	i
!	20	/
!		
·		
!!		
JMAT Index or 0	7×n	
Jini Lidex of V	7-11	}
j JIT DST Number or 0		i
New Feb. Balance A		!
Next Entry Pointer or 0		- Ldev n
Classname under which LDEV is associated or undefined.		
		i

G.00.00 15- 13

### Sysdump/Initial/Store

## CHAPTER 16 SYSDUMP/INITIRE

# CONFORTA File

# Record O of CONFORTR File (CTABO)

1100.00				
0		  0		
1	CURRENT VERSION OF CTRB	1		
2	STRNDARD STACK SIZE	2		
3	CORESIZE IN K WOROS	3		
4	TERMINAL BOUND PRIORITY	4		
5	NORMRL PRIORITY	5		
6	CPU BOUND PRIORITY	6		
7	# OF SECONOS TO LOG-ON	7		
10	LOG FILE RECORD SIZE (SECTORS)	8		
11	LDG FILE SIZE (RECORDS)	9		
12	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	10		
13	LOG BITS (ONLY 11 USED)	11		
14 15 16	< <defines being="" is="" ldgged="" whrt="">&gt;</defines>	12 13 14		
17		15		
20	DEFRULT JOB/SESSION CPU TIME LIMIT	16		
34	MRXIMUM OPEN SPOOL FILES	28		
35	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	29		
36	MAXINUM # OF SPOOL FILES (KILD SECTORS)	30		
37		31		
40	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	32		
41	# SECTORS PER SPOOL EXTENT	33		

#### G.00.00 16- 1

### Sysdump/Initial/Store

# Record 1 of CONFDATA File (CTAB)

0		ļ
1	# OF DST ENTRIES	1
2	# OF PCB ENTRIES	2
3	# OF IOQ ENTRIES	3
4	# OF TERMINAL BUFFERS	4
5	# OF CST EXTENSION ENTRIES	5
6	INTERRUPT CONTROL STRCK SIZE (Q1 to Z1)	6
7	# UCOP REQUEST QUEUE ENTRIES	7
10	# BREAKPOINT ENTRIES	8
11	# TRL ENTRIES	9
12	# LOCAL RINS	10
13	# GLOBAL RINS	11
14	# OF SYSTEM BUFFERS	12
15	# OF CONCURRENT PROGS	13
16	LORGER SEGMENT SIZE	14
	######################################	
24	SIZE OF VIRTUAL MEMORY	20
25	OIRECTORY SIZE (SECTORS)	  21
		:

G.00.00 16- 2

# Sysdump/Initial/Store

# CONDRIA (Cont.)

36	MRXIMUM CODE SEGMENT SIZE	30
37	MRXIMUM # OF CODE SEGMENTS/PROCESS	31
40	MAXIMUM STRCK SIZE (MRXDATR)	32
41	MAXIMUM EXTRA DATA SEGMENT SIZE	33
	MRXIMUM # OF EXTRE ORTA SEGMENTS/PROCESS	34
50	MRXIMUM # RUNNING SESSIONS	40
51	MRXIMUM # OF RUNNING JOBS	41
52	# LDG PROCS	42
53	LDG 10's	43
54	# OISC REQUEST TRBLE ENTRIES	44
55	# SPECIAL REQUEST TABLE ENTRIES	45
56	# PRIMRRY MESSAGE TRBLE ENTRIES	46
57	# SHRP TABLE ENTRIES	47
58	# SECONDARY MESSAGE TRBLE ENTRIES	48

# Sysdump/Initial/Store

## OEVORTR. PUB. SYS

# Overview

PARAMETERS
DRIVER TABLE
LPDT
LDT
LOTX
CLASS/TERM HERDER
CLASS
TERM DEF
ROD'L DVR TRBLE
CS DEF
CS TABLE

# Parameter Record

0	CNECXSUM
1	VERSION
2	NEXT RECORD
3	HIGHEST LDEV
4	HIGHEST DRT
5	NR. RDD'L DRIVERS

### Sysdump/Initial/Store

64	REC #	DVR TABLE
	LENGTN	
66	REC W	LPOT
	LENGTH	
68	REC #	LDT
	LENGTH	l
70	REC #	LDTX
	LENGTH	
72	REC #	OCTN
	LENGTH	,
74	REC #	CLASS
	LENGTH	
76	REC #	TERM DEF
	LENGTN	
78	REC #	ADD'L DVR
	LENGTH	
80	REC #	CS DEF
	LENGTH	
82	REC #	CS TABLE
	LENGTH	

Sysdumg/Initial/Store

## Oriver Table

0 1 2 3 4 5 6 7	TYPICAL ENTRY FORMAT	
D	Į R	
I	V	
N N	A	
H	E	

OS OS OEVICE (if set ORT is zero)
CR CORR RESIDENT
CHAN H CHRNHEL
HRSTER LDEV LDEV of device which this DS device is linked to.

Words 3-7 contain the driver name.

G.00.00 16- 5

G.00.00 16- 6

# Sysdump/Initial/Store

# SYSDUMP Format

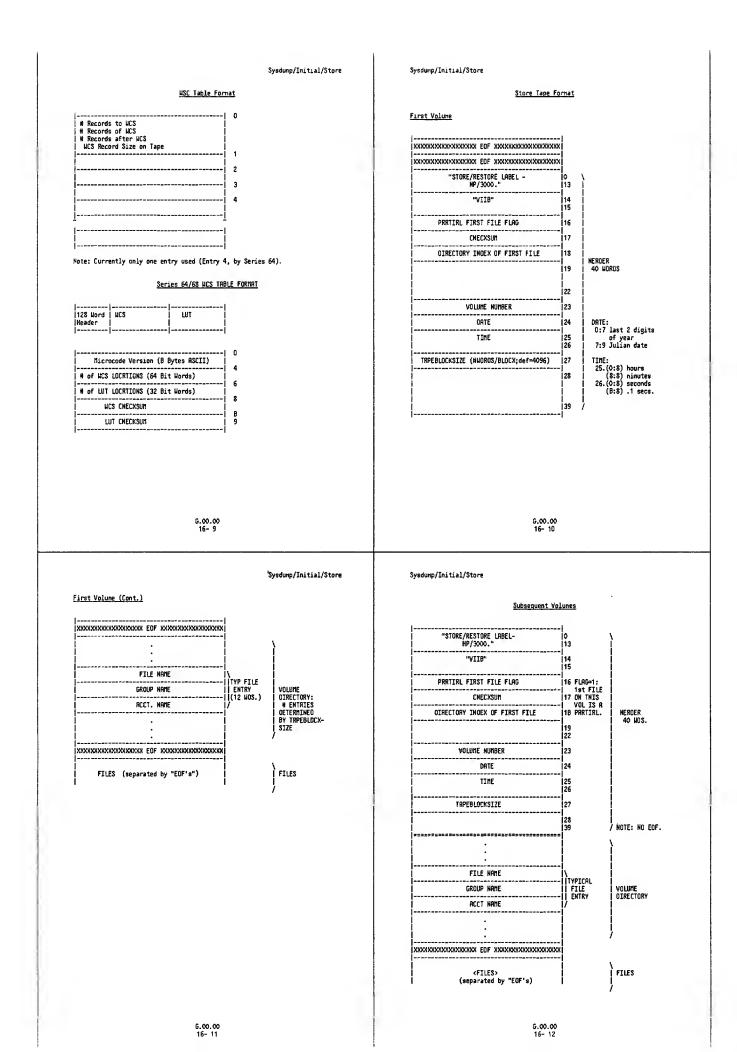
	AMIGO CHANNEL PROGRAM	<pre><entry #1="" (rom="" baseo<="" point="" td=""></entry></pre>
i	AMIGO	127
->	NCS TABLE	
	NCS #1	4.0
	UCS #2	Only for the 64/68. Refer to the NCS Table for the 64/68 below.
	UCS #n	
	CHECXSUM Anigo	<pre><entry #2="" (hcs="" 0="" 127<="" based="" machines)="" pdint="" pre=""></entry></pre>
	AMIGO	161
	ICS	
	LOW CORE	
	Initial CST	
	CS TABLE	
	DEVICE CLASS TABLE NERGER	
	DEVICE CLASS TABLE	
	TERMINAL DESCRIPTOR TABLE	
	VTAB	
	OLDVTRB	*
	OISC COLD LOAD INFORMATION TABLE	*
	CTAB	
	CTABO	
	COMMUNICATION RECORD	
	CSDVR	
	CSOEF	
	INITIAL'S DB AREA	
	G.00.00 16- 7	I

# Sysdump/Initial/Store

ļ	STACX MARXER	
	ORIVER TABLE	
	LPOT	
	LDT	
ļ	хтд	
	INITIRL'S SEGMENTS	
ļ	RIN TABLE	*
	LOGGING IDENTIFIER TABLE	*
	OIRECTORY MEAGER	*
	, DIRECTORY	*
	NXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	
	SYSTEM PROGRAMS, SL, NON-STO. ORIVERS	
	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	
	STORE/RESTORE HEADER	
	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	
	STORE/RESTORE DIRECTORY	*
	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	
	USER FILES (SEPARATEO BY "EOF's"	*
	STORE/RESTORE TRAILER	
	XXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXX	
	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	
	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	

\* NOT DUMPEO IF DATE = CRRRIAGE RETURN

MOTE: ON OISC, READ-SIO-PROGRAM KEPT IN OISC LABEL.



# End of Volume

		<b>Y</b>
<files> (separated by "EOF's)</files>		FILES
DXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		′
"STORE/RESTORE LABEL-NP/3000."	  0  13	}
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	14	!
	20	ļ
FLAG: PRECEDING EOF MARXS FILE ENDED	21	TRAILER
FLAG: PRECEDING EDF MARKS TAPESET ENDED	22	40 MDS
VOLUME NO.	23	
ORTE	24	
TINE	25 26	ļ
	27	
	39	7
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	İ	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		

G.00.00 16- 13

#### Labsisd Tape Subsystem

#### CHAPTER 17 MISCELLANEOUS

## Labeled Tape Subsystem

The MPE labeled taps subsystem permits convenient access to tapss labeled to either RMSI or IBH standards. It operates as a set of subprocedures to the file system. R labeled taps consists of one or more logical files. Each logical file consists of three physical files, i. s. taps areas delinited by tappmarks. The first physical file contains header labels, the second contains that data, and the third contains trailer labels which are (except forminor differences) copies of the header labels. The taps mark following trailer labels will be followed sither by header labele for the next file, or by another tappmark if there is no next file. Labels are 30 bytes long, and conventionally are identified by their first four characters (three letters and a digit) and contain information as follows (CP:= character position; L:= length):

VOL1: Present only on the first file of a volume, the volume label contains the volume identifier, which is usually the number on the taps strap, and is thus not expected to be changed.

CP	Field Name	L	Content
1/3	Label identifier	3	"YOL"
4	Label Mumber	1	"{"
5/10	Volume Identifier	6	Vol ID
11	Accessibility	1	"O" if IBM, slss " "
12/79	Not used	62	81anks
80	Label-Standard Version	1	"1" if HP RMSI else " "

UVLn: User volume labels. May be present on tapes from foreign shops, but are not unitten by NPE. If encountered, they are ignored.

G.∞.∞ 17- 1

### Labeled Tape Subsystem

ļ		!	format).
11/15	Record Length	5	Record length (adhering to to RPE rules) in charactere.
16/23	Lockword	8	MPE File Lockword.
24/36	Not Used	13	MPE writes blanke
37	Record Typs	1	"A" = ASCII "B" = Binary.
38	Carriage Control	1	"C" = control " " = ne control.
39/80	Not Used	42	81anke

# IBN has a slightly different format. It is:

CP	Field Name	L	Content
1/3	Label identifier	3	"HOR"
4	Label Number	1	"2"
5	Record format	1	"F" = Fixed "V" = Variable "U" = Undefined Others treated as Undefined
6/10	Block Length	5	Block length (in character format).
11/15	Record Length	5	Record length (adhering to to MPE rulss) in characters.
16	Nat Used	1	Blank.
17	IBM Position	1	"O" = no volume suitch "1" = a suitch has occurred.
18/38	Not Used	11	Blanke.
35	IBM Block Attribute.	1 1	"B" = Blocked records. "5" = Spanned records. "R" = Blocked and Spanned. "" = No blocked or spanned.
40/80	Nat Used	41	Blanks

#### Labsled Tape Subsystan

HDR1: First header label. Required for each file. Specifies:

CP	*			
4 Label Number 1 ""  5/21 File Identifier 17 File name, if tape was not written by nPE, only the first eight are significant.  22/27 Volume Set Identifier 6 Names the volume on which the set of files begins 128/31 Real Number 4 Counts the reals that contain this file (1 starts) 132/35 file sequence number 4 Counts the file in the set of files (1 starte) 136/41 Not Used 6 NPE writtes blanks 142/47 Creation Date 6 Vear and day within year when the file we written 148/53 Expiration Date 6 Vear and day within year within the file was proved written without permission.  54 Roceseibility 1 X230 if Lockword, "O" if IBN 155/50 Block count 6 Number of blocke if IBN. 161/73 System Code 13 "NP MPE 3000"	CP	field Name	ļ L	Content
S/21   File Identifier   17   File name, if tape was not written by APE, only the first eight are significant.     22/27   Volume Set Identifier   6   Names the volume on which the set of files begins     28/31   Real Number   4   Counts the reals that contain this file (1 starts)     32/35   file sequence number   4   Counts the file in the set of files (1 starts)     32/35   file sequence number   4   Counts the file in the set of files (1 starte)     32/35   file sequence number   4   Counts the file are in the set of files (1 starte)     36/41   Not Used   6   NPE writtes blanks     42/47   Creation Date   6   NPE writtes blanks     48/53   Expiration Date   6   Vaar and day within year when the file has written without permission.     54   Rocsseiblity   1   X200   If Lockword, "O" if IBN     55/60   Block count   6   Number of blocke if IBN     61/73   Systen Code   13   "NP IPE 3000 "	1/3	labsl identifisr	3	"HDR"
S/21   File Identifier   17   written by APE, only the first eight are significant.	4	Label Number	1	"1"
ths set of files begins  28/31 Resl Number  4 Counts the resis that contain this file (1 starts)  32/35 file sequence number  4 Counts the file in the set of files (1 starts)  36/41 Not Used  6 NPE writes blanks  42/47 Creation Date  6 Year and day uithin year when the file was uritten.  7 Year and day uithin year when the file was uritten without permission.  54 Roceseibility  1 X230 if Lockword, "O" if IBM  55/60 Block count  6 Number of blocke if IBM.  61/73 Systen Code  13 "NP MPE 3000"	5/21	File Identifier	17	written by MPE, only the
contain this file (1 starts)  32/35 file sequence number   4   Counts the files in the set of files (1 starte)  36/41 Not Used   6   NPE writes blanks    42/47 Creation Date   6   Vear and day within year when the file was written within file was written within the file was written such that the file was proved written such out provision.  54 Recessibility   1   X230 if Lockword, "O" if IBN    55/60   Block count   6   Number of blocke if IBN    61/73   Systen Code   13   "NP IPE 3000 "	22/27	Volume Set Identifier	6	
Of files (1 starte)     36/41   Not Used   6   NPE writes blanks     42/47   Creation Date   6   Year and day within year when the file was written.     48/53   Expiration Date   6   Wear and day within year when the file has be overwarded in the file has be overwarded in the file has be overwarded in the file has be overwarded in the file has be overwarded in the file has been day of the file has been day of the file has been day overwarded in the file has been day overwarded in the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been day over the file has been da	28/31	Resi Number	4	
42/47   Creation Date   6   Year and day within year when the file was written	32/35	fils sequence number	4	
when the file use written.	36/41	Not Used	6	MPE writes blanks
43/53   Expiration Date   6   when the file may be over- written without permission.   54   Recessibility   1   X230 if Lockword, "O" if IBM	42/47	Creation Date	6	
55/50   Block count   6   Mumber of blocke if IBM.   61/73   System Code   13 "NP MPE 3000 "	48/53	Expiration Date	6	when the file may be over-
61/73   System Code   13   "NP HPE 3000 "	54	Rocsseibility	1	X230 if Lockword, "O" if IBM
	55/60	Block count	6	Number of blocke if IBM.
74/80   Not Used   7   Slanks	61/73	Systen Code	13	"NP MPE 3000 "
	74/80	Not Used	7	81anks

 $\ensuremath{\mathsf{hDR2}}\xspace$  . Second header label. Although defined by the standard, may be missing on foreign tapse. Contains:

CP	Field Name	ļ L	Content
1/3	Labsl identifier	3	"HOR"
4	Label Number	1	"2"
5	Record Format	1	"F" = Fixed   "Y" = Variable   "U" = Undefined   Others treated as Undefined
6/10	Black Length	5	Block length (in character

5.00.00 17- 2

#### Labsied Tape Subsystem

User header labels: optional. Standard prescribes UHLn in the first four characters, but HPE doesn't care.

ECV1: End of Volume; used as first trailer label. Required if the logical file is continued onto another real. Identical to MDR1, except contains the number of physical blocks of data in the data area.

ÇP	Field Name	L	Content
1/3	Labsl identifier	3	"EOV"
4	Labsl Number	1	"1"
5/54	Sans as HDR1	50	
55/60	Block Count	6	Number of data blocks sincs last beginning of fils section label group.
61/80	Sans as HDR1	20	

ECV2: Defined by the standard, but may be missing on foreign tapee. Follows ECV1: format same as  ${\rm HOR2}.$ 

EOF1: End of file; used as first trailsr label. Required if this is the end of the logical file. Format same as EOV1.

EOF2: Sams as EOV2 sxcept used after EOF1.

User trailer labeler optional. Standard prescribes UTLn in the first four characters, but MPE again doesn't care.

#### Tape Label Table

The tape label table is the private playground of the tape label subsystem. It consists of two parts: LDEV Control Blocke (LGBs) and Volume Control Blocke (VGBs). The LDEV area is set up at system initialization and contains one entry for each magnetic tape LDEV and serial disc device in the system. Rs is common in TPE, the first entry is a dummy which tells where the other things in the table are. The volume area contains one entry for each labeled tape volume requested or active on the system.

Rithough table entries are stored in an extra data segment, they are generally manipulated via local copies on the stack. The procedures GETLDEY and GETFMUN look for LDEV and volume entries as specified; they copy then to etack buffere and return the DST address for use in copying then back. PDSTVTEMT copies the entries back, and in the case of a new volume entry, allocates apace for it in the volume section of the tape label table.

Initial will build the "uminitialized" TLT as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
<u> </u>	\$1	ze 0	ft	tal	ile,	10	uord	s (s	Luay	*	1)				_	0
	Nu	nber	of.	LDEV	111	the	tab	le s	×							1
fla	g=1					LDE	V#							. <b></b> .	17	2
			Tot	al o	F LD	EVS	(X)	entr	162	of .	abov	e			i	
						LDE	V)								17	X+2
 				ns10 r1ng			APES	: 							ا ا	

I: 1 if Tage drive O if not Tage drive (i.e. serial diec)

6.00.00

#### Labeled Tape Subsystem

During PROGEN, SETUP'TRPES is called to initialize the table. The overall structure of the initialized TLT is:

TLTDST -- 232,#26 TLTSIR -- 247,#39

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Table initialization word (=1 when initialized) 1 0 Entry size (ESIZE) = 232,#26 2 Table relative pointer to base of LCB entries (LTBRSE) (1) Table relative pointer to base of VCB entries (VTBRSE) (2) 3 Table relative pointer to top of Volume table (VTTOP) (3) Size of Tape Label Table, in words (VTMRX) 5 6 10

not used

j	30
	31
	32 (-(1)
LDEV Control Block area one entry/mag tape drive	\-(\)
1	
	<-{2}
	i (2)
and free entries	
	<-(3)
Area available for expansion of VCB table	'
I.	1

-----

## Labeled Tape Subsystem

The LCB entries have the following structure:

	0 1 2 3 4 5 6 7 8 3 10 11 12 13 14 13	
į	Type   T   L   B   HP	0
-	Logical device number	1
-	VCB address	2
-	Reel number	3
ŀ	File sequence number	4
ŀ	Ereation date	5
ŀ	EXPLICATION COLE	6
ľ		7
ļ	File name	10
١		1
ı		1 16
Ì	<b>+</b>	1
į		j 17 +
į		50
į	(not used)	21
	(1101 0820)	22
1		23
1		24
	Volume set identifier	25
		26
		27
	Volume identifier	3
		3
		۱-

#### Labeled Tape Subsystem

Type: 00 = no tape nounted 01 = unlabelled 10 = RMSI 11 = IBM L: 1 if file has lockword.

IT 11 fate mas abbonum.

T: 1 if device is a tape drive.

B: 1 if tape is from Burnoughs, which has incorrect block/record size in the HDR2 label. Code can be patched to correct the size.

HP: 1 if tape is Hewlett-Packard RHSI format.

VCB address: Pointer to VCB entry describing volume mounted on tape drive, only if linked. Otherwise, O. The VCB formatis:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

A   F   D   Position   W   SeqTyp  LblTyp  L   T   R   B	0
₩ WEV #	1
PIN	2
File number (AFT index)	3
File sequence number	4
S   R   D   C   Density   V   Reel number	5
Expiration date	6
	7
File name	10
'	
!	16
ļ	17
4	+ 17
!	20
Lockword	21
I FOCKHOLD	22
	23
	24
Valume set identifier	25
	í

#### Labeled Tape Subsystem

#### VEB (Cont.)

	26
	27
Yolune name	30
	31

R: ASCII FORTION
F: Flush bit - operator did REPLY 'pin>,0.
0: DEVREC Wait (used with reelsuitching).
Position: Gives head position within logical file.
0 = at load point (LOPNT)
1 = HORI label next (NINX)
3 = after HORE label (RN2)
4 = after user header labels (RNU)
6 = data next (ONX)
7 = after data (RD)
8 = EOFI/EOVI label next (TINX)
10 = after EOF2/EOV2 label (RT2)
11 = after user trailer labels (RTU)
W: Wirls access specified.
SeqTyp: File open eequencing type.
0 = natch filename
1 = NEXT
2 = RODF
3 = use file sequence number

2 = RODF

3 = use file sequence number

LblTyp: Re in LCB entry.
L: Linkwait - mark left by CHEATETLIENT for LINKLABEL.
Ti. Hount wait - waiting for operator to nount tape on FOPEN.
R: Reelswitch wait - waiting for next reel.
B: Busy bit - this entry is in use.

LDEV W: Logical device runber of tape drive with this volume, only if linked. Otherwise, 0.

S: SIDRE tape.
R: REELSUITCH has been done. Used by SIDRE/RESIDRE to handle SIDRE label and directory file.
C: Mext file is directory. Used by SIDRE.
C: VOL1 label is to be created (uritten).
Density: volume set density. During a volume set open, contains the density requested by the user in FOPEN. Once the volume set is open, contains the actual density of the volume set. Only valid for tapes on variable density tape drives.

O = default density for volume set open
1 = 1500 BPI
2 = 6250 BPI
V: 1 if volume set is being opened. Reset after completion of FOPEN.

G.00.00

## Labeled Tape Subsystem

(including user labels) is maintained. There is a separate CRSE leg for each

If an EOI reflective mark or an EOF in data ie found, REELSUITCH is called (principally from the file system procedure IOROVE) to call for the next reel, if any. If another reel is needed, the tape drive is set Unounced so that ROREC will be called to recognize the new tape when it is nounted. REELSUITCH returns to its caller when it is satisfied that an appropriate table is mounted.

#### Closing Files

FCLOSE calls CMECKUL to handle uriting EOF1 and EOF2 if needed and resolving the tape position. If the disposition is 3, the tape is left positioned at the next file. If the disposition is 2, the tape is supposed to be left at the beginning of the current file, but the code does not presently provide for reelsuitching if the present file began on a prior reel.

At present, ensuing volumes of a multi-volume cet must be nounted on the came drive as the first, mostly because neither the file system nor STORE-RESTORE was capable of dealing with LDEV changes in the middle of a file. REELSHITCH reports the LDEV being used, housever, eo that the capability of using a different LDEV can be added in the future.

#### Store-Restore

Complications ensue on labeled STORE-RESTORE tapes because there neede to be a file directory at or near the beginning of each tape of a multi-volume eet; RESTORE usee this directory to determine whether the specified file(e) can exist on this tape. Secause the real euttehing process would otherwise be invisible to STORE-RESTORE, special bits (VCB-RSUDOME and VCB-WRITOIR) are kept to enable special intrinexis callable by STORE-RESTORE to report whether a directory neede to be written or is about to be encountered.

The special procedure MEXITAPEFILE is used by SIGNE-RESIGNE in lieu of doing a FCLOSE(.3) followed by an FODEN to get to the next file. This permits cleaner handling of both REPLY O and Forward Space (logical) File over a Reelswitch, as well as saving the time needed to tear down and reconstruct all the control blocks.

#### Miscellaneous

PYOLID is used by the SHDWDEV command processor (in SPOOLCOMS) to obtain the name of the volume on the specified drive uithout having to know the structure of the tape label table. For the same reason, IGETIMFO ie used by the FFILEIMFO intrinsic (in FILEIO) to get labeled tape information.

System failure 86 in TPE is defined as a major problem in LRBSEG. Generally speaking it is a problem with the TLT setup, for example if LABSEG cannot find an LDEV in the table.

#### tabeled Tabe Subsystem

#### Volume Recognition

Volume recognition is the responsibility of DEVREC, which reads the first record of a newly-nounted tape on an undered drive and passes the record to RVREC. GVREC may see: VBul in the first 4 bytes, in RSCII, in which case the tape is RSI; VBUI in the first 4 bytes, in CBCDIT, in which case the tape is RSI; Mnything else, in which case the tape is considered unlabelled.

If the tape is unlabelled, RVREC reports to DEVREC that no further action is required. If the tape is labelled, RVREC wants to see the first MORI label, so asso DEVREC to read another record. (Unfortunately, DEVREC cannot be etopped long enough for RVREC to do its own read.) When the MORI record is found, the volume entries can be searched to eee if there is a pending request for this volume. If so, the waiting process is restarted.

If the system has been restarted with tapes mounted, there will not be interrupts to alert DEVREC. The procedure RECOGNIZE is called when needed to see if any such tapes exist.

## Opening a File

FOPEN gets into the tape label code in three different places. The first is to call CRERTETIENT, which parses the string passed in the FORMSMSD parameter to identify the labeled tape file required. If there is no existing corresponding entry in the volume area, thie is a volume set open, and a new volume entry is created. There hay be an existing entry (if the tape use FOPENed and FCLOSEG with disposition 2 or 3), in which case there is an aesociated LOEV entry for the drive on which the tape was left nounted by the prior operation; in this case, the new information is stuffed into the existing volume entry. R bit (LINKURII) is left set to mark the entry for LINKLABEL.

The second entry is through LINKLABEL, which is called from GLIOCRTE. Rt this time, it is necessary to identify the LDEV to be used for the tape. If no LDEV is associated, the LDEV entries are searched to see if the operator has already nounted the required tape; if so, the volume and LDEV entries are cross-tied and LINKLABEL is done. If the search turns up nothing suitable, the operator is requested to nount the appropriate tape, and the procedure waits for either a REDY or for AVREC to discover the appearance of a suitable tape and restart the process. If the operator enters a reply, it is validated. validated.

The third entry is through POSITION, which is responsible for positioning the tape to the requested file. At the file, the MORI and MORZ label are examined as required to determine the file characteristics.

### Reading and Writing Files

All procedures which move tape go through the catchall procedure CHECKUL, which takes care of necessary labeled tape doings. The code insuree that the sequence: header labele (including user labels), data, trailer labels

#### Breakpoint Table

#### Breakpoint Table

DST = 30(10) = X36

The break point table is divided into 2 sections:

- PCB BRERKPOINT EXTENSION TABLE (PCB'8KPI'EXT)
   This table contains the heads of the breakpoint
- 2) BRERKPOINT ENTRY TPBLE (BKPT'ENTRY'TRB)
  This table contains the actual entries

General Layout

PC8(18)	PCB'8KPT'EXT
	>
	-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-
SYS GLOBAL 14:15 x25   :L:SI	
<pre>L = fable locked S = System break     points exist</pre>	
	j 

### Breakpoint Table

### PCB Breakpoint Extension Table

I # ENTRIES	ł	ENTRY SIZE =	1
I HEAD SYSTEM LIST	Ī	FREE ENTRY =	0
! N USED USER ENTRIES	1	RCTIVE ENTRY =	Index 1st Entry
USER ENTRIES	ī		chain in breakpoint
1	1		
***************************************			

# Breakpoint Entry Table

	ENTRY (0)	FREE ENTRY	
0	I# WORDS BRERKPOINT TAB	11: SIZE	Ī
1	HERO FREE LIST	FORWARD LINK	<b>-</b>
2	# MORD USED	BACKWARD LINK	ī
3	MAX N WORD USED	!	1
4-6	UNUSED	4	1
	LAST ENTRY		ļ
0	1		

The breakpoint entry table consists of variable length entries The minimum entry size is  $7.\,$ 

#### Breakpoint Table

#### Active Entry

0 1:2:3 4:5:6 7:8:9 0:1:2 3:	4:5
0 P:L:V D:F:T U:P:C U: SIZE	
M  UNUSED	 
BLOCKLABEL	<del>-</del>
PLOC	1
I INSTRUCTION	1
FINK	1
I USERLABEL	ī.
CONDITION/COUNT	, satrapje
ı	1 .
COND DESCRIPTOR	·
	O   P:L:VID:F:T U:P:C U: SIZE     : :   : :   : :   P:

G.00.00 17-13

5.00.00 17- 14

#### Breakpoint Table

# Breakpoint Entry Table (Cont.)

```
FREE ENTRY

1 = FREE
0 = USEO
0 = USEO
PRIVILECED HODE BREAKPDINT
1 = PRIV.
0 = NON-PRIV
PROCESS-LOCKL BREAKPDINT
1 = PRIV.
1 = NON-PRIV
PROCESS-LOCKL BREAKPDINT
1 = PRIV.
1 = PRECESS-LOCKL
0 = SYSTEM
VRILORITION BIT
1 = INSTRUCTION HOT IN TAB.
DOUBLE IRAP
1 = BREAKPDINT OSCILLATES BETWEEN
P/P+1
0 = NOT DOUBLE TRAP
FAKE 'DUTHY' TRAP
1 = BREAKPDINT AT P+1
0 = BREAKPDINT AT P+1
0 = BREAKPDINT AT P+1
0 = BREAKPDINT AT P+1
0 = BREAKPDINT AT P+1
1 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = TRAP TO USER SUPPLIED LABEL
0 = RODOJ COUNT
UPDRITING
1 = ENTRY IN PROCESS OF BEING
UPDATED/REMOVED
0 = NOT DEZING UPDATED/REMOVED
USER PLABEL MODE
LINK
0 = END OF CHAIN
>0 = END OF CHAIN
>0 = END OF CHAIN
>0 = END OF CHAIN
  ENTRY(0). (0:1) = FR:
  ENTRY(0).(1:1) = P:
 ENTRY(0).(2:1) * L:
 ENTRY(0).(3:1) = V:
 ENTRY(0). (4:1) = D:
 ENTRY(0).(5:1) = F:
 ENTRY(0).(6:1) = T:
 ENTRY(0).(7:1) = U:
 ENTRY(0).(8:1) = PH:
 ENTRY(0). (9:1) = C:
 ENTRY(0).(10:1) = UP:
ENTRY(1).(0:1) = M
ENTRY(6) = LINX:
```

#### Breakpoint Table

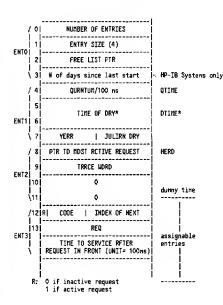
## Breakpoint Entry Table (Cont.)

		-			
CI	JUNT			CONDITION	
1)   ORIG	INAL CNT.	i		OPERRNO1	1
H OF	HITS	ł		OPERAND2	1
1		1	I OP	11 0Pt2  RELOP	1
RELOP -> (8	3 = 4 = 5 =	ET EQ	9 = LTI 10 = GTI 11 = NE	<u> </u>	
OPT1 -> (0 OPT2 -> (2	:2) OPER ::2) OPER	AND1'S T AND2'S T	YPE		
1 -> AD	ES: HSTANT ( DRESS (D DIRECT R	CUBLE NO	RD)	iORO)	
JPERRHÓ FOR Constrní	->	DMST	<u>.</u>		
ADDRESS	RI	G   BRS DFFSET ). OFFSE	   	PE 3 ONLY)	
REG	-> (0:6	3 = 4 = 7 = 1	A 10 Sy 11 Dr 12 Dk 17	= Q = S	REGY":
BASE	-> (6:1	O) SEG	N/BRNK N		

#### Breakpoint Table

### Timer Request List (TRL)

The system clock interrupts every 100 ms, with the CR being autonatically cleared. Rn exception is the Shared Clock Interface measurement service which allows rates as fast as 5 ms. The interrupt handler is the procedure IICK. On entry, DB is pointing to the base of timer request list. Besides timeout requests, the clock also controls time slicing.



G.00.00

#### ' Timer Request List

#### MPE User Logging

MPE USER LOGGING enables users and subsystems to log changes to data sets on disc or serial files. This "change" file can later be used to recover data lost due to a system or program failure. The log file can itself be used for auditing purposes.

#### General Design Overview

## Hardware Environment

No special hardware is required to operate the system. However, if logging to a tape file is desired, the hardware configuration must include a tape drive. If there is no tape drive, then may log to a serial disc class device.

# Software Environment

MPE User Logging is an integral part of MPE. No other special software is required.

## Design Narrative

User Logging enables users and subsystems to journalise additions and modifications to HPE and subsystem files. The journal can reside on either disc or serial logfules.

User Logging consists of a logging process, a menory buffer, a disc resident logging buffer (for serial logging) and a user defined destination log file on disc or serial neda:

The logging process has two functions depending on whether the destination file resides on disc or serial media. If the destination file is serial, the logging process performs all output to the destination file. If the destination file is on disc, the logging process allocates additional space (extents) as it is required by the user.

The logging buffer is divided into communication and buffer areas. The communication area is used to pass information among the users and the logging process. This information includes status of the logging process and logging file, space remaining in the logging file and error information important to users or the logging process. The buffer portion of the logging data segment blocks inputs into the logging file before the data is actually posted. The buffer is flushed any time a user requests to close a log file or when a logging process is terminated. (The buffer is also flushed by the begin/end transaction or buffer flush requests).

#### Timer Request List

#### TRL (Cont.)

CDDE	& REQ	indicate the type of	request.
	CODE:	REQ:	TYPE:
	٥	DITP	Hangup
	1	DITP	Carrier failure
	2	DITP	202 turnaround
	3	DITP	Read
	4	DITP	Logon
	5	PC88 index	Delay
		to process	
	6	DITP	LP not ready
	7	DITP	2640
	<b>X10</b>	Port Hask	Msg port timeout
	<b>X11</b>	DITP	Block mode read
			timeout (30 secs)
	712	PCBB index	Watchdog timer for
		to process	process

The list of pending requests is kept ordered by time with later entries at the tail.

X20-X37	DITP	SIO device timeout: DIT8. (code_1 on expiration, cleared on Tinereg.
<b>25/26</b>	*DTIME	For Series 30/33, DTIME is # of TICS (0.091457 ms) since last midnight.

G.00.00

Timer Request List

#### Error Recovery Description

The error recovery mechanisms provided by User Logging are: power fail recovery and recovery from system failure.

Power failure recovery applies only to tape log files since MPE provides adequate recovery for disc files during power fail. When a power failure is detected, a message will be printed on the console asking the operator los place the tape drive back on-line. (If the operator places the tape on-line before the message valid data may be overwritten). (To reset the tape drive the operator must hit the load button until the tension returns to the drive. Then hit the reset button followed by placing the tape drive back on-line). At this time the log process will recover the file by rewinding to the load point and then forward spacing to the point where the power fail occurred. Writing to the log file will continue at that point.

In the event of a system failure, the warm start load option initiates recovery of User Logging files. In the case of a serial file, the file is read and compared to the disc logging buffer. Ril records found in the disc buffer that are not on the serial log file are posted and a proper end of file written. If the destination file is a disc file, all records are read and verified and an end of file posted to the file. In order to continue logging to a User Logging file that has been recovered in this manner, the logging process for the file must be restarted using the console command: LOG.

HOTE:

Rny records in the buffer area of the logging buffer will be lost.

User logging has been enhanced to work with labeled serial discs. Intermally the log process handles serial disc (or cartridge tape) log files the same as for tape files.

User Logging Table

# Design Structures

#### User Logging Table

ENTRY SIZE = N38 words DST X33

Table containing an entry for each activated user logging process. Each entry is created when the process is started, and deleted when the process terminates. (Via : LIOC command). The information is extracted from the logging Identifier Table (LIDTAB).

	ENTRY O	
N		x
۰ .	NUMBER OF ENTRIES	0
1	FREE ENTRY HEAD PT.	1
2	INUSE ENTRY NEAD PT.	2
3	NEXT BUFFER NUMBER	3
4	MAX N PROCESSES	4
5	MAX N USEAS/PAOCESS	5
6		6
7	ENTRY SIZE	7
37	:	45

### WORD ENTRIES

NUMENTRIES	=	LOGTAB
FREE	=	LDGTAB(1)
INUSE	=	LOGTAB(2)
BUFNUM	=	LOGTAB(3)
MAXLOGPACC	=	LDGTAB(4)
MAX'USA'PAOC	=	LOGTAB(5)
INGTOR'ESTAE	=	INGTER(7)

G.00.00 17- 21

User Logging Table

NUMENTAIES
The number of entries in the logging table.

FREE R table relative pointer to the first free entry in the logging table. (-1 = table full).

INUSE R table relative pointer to the first entry in the logging table that is being used (-1 = no entries in use).

BUFNUM The number of the buffer associated with this logging process. Used to create the name of buffer file if serial logfile. (i.e. ULDGxxxxx.PUB.SYS).

MAXLOGPROC
The naxinum number of user logging processes allowed.

MAX'USA'PAOC The maximum number of users per logging process.

LOGIAB'ESIZE
The size (in words) of each entry in the table.

G.00.00 17- 22

User Logging Table

Typical Entry		x
0	LOGGING	- 0
	i_ IDENTIFIER	-
	-	-
4	<u> </u>	-¦ 4
	i_   BUFFEA	-
	- NAME	-
	ļ_	-
8	<u> </u>	_  <sub>10</sub>
-	i_  - FILE	-
	I NAME	- <u>İ</u>
	-	-
12		—i 14
	LOCX	-  ''
	- HORD	-
	-	4
16		
	. GROUP	-  ``
	-	-
	_	_
20		_ 24
20	- ACCT	_
	-	-
	ļ <u>-</u>	_
24	NUMBER OF USERS	_  _,
24 25	BUFFER DST NO	
	i	31
26	LOG STATUS	32

G.00.00 17- 23

User Logging Table

27	CUAR RUTO   CURR TYPE	33
28	LOG DEV	34
29	LOG PCB N	35
30	SHITCH FLAG	36
31	NEW AUTO NEW TYPE	37
32	ROORESS OF	40
	LOGGING BUFFEN	
34	SIZE OF	42
	LOGGING BUFFEN	
36	FUAD ENTRY PT	44
37	BUAD ENTRY PT	45

TABINOEX BTABINDEX DTABINDEX WOAD INDEX TO CURRENT ENTRY BYTE INDEX TO CURRENT ENTRY DOUBLE INDEX TO CURRENT ENTRY LGNAME BNAME LFNAME LFLOCKH LFGROUP LFRCCT BTABINDEX BTABINDEX+8 BTABINDEX+16 BTABINDEX+24 BTABINDEX+32 BTABINDEX+40 BINDLW-A-40
HRBINDEX-24
HRBINDEX-25
TRBINDEX-26
TRBINDEX-27. (0:8)
HRBINDEX-27. (0:8)
HRBINDEX-27. (0:8)
HRBINDEX-30
HRBINDEX-30
HRBINDEX-31. (0:8)
HRBINDEX-31. (0:8)
HRBINDEX-31
HRBINDEX-36
HRBINDEX-36
HRBINDEX-36
HRBINDEX-36
HRBINDEX-36
HRBINDEX-36
HRBINDEX-37 NUMUSERS NUMUSERS
DST
STATUS
LGRUTO
LGTYPE
LGDEV
PIN
LGSLITCN
LGNEHRUTD
LGRUTYPE
LGRUTOR
BSIZE
NEXT
PREV

User Logging Table

 $\ensuremath{\mathsf{LGNAME}}$  The name of the logging process (logging identifier).

BNRME
The name of the disc buffer used if the logging process destination file is a sexial file. This is a file that resides in PUB.SYS. The format of the name is ULOGXXXX where XXXX is the buffer number padded on the left with zeros.

If the switch fl/g is true, the following will be the fully qualified file name of the new log file.

LENAME
The name of the logging file.

LFLDCKU

The lockword of the disc logging file.

The group that the destination logging file resides in if the file is a disc file.

The account that the destination logging file resides in if the file is a disc file.

 $\ensuremath{\mathsf{NUMUSERS}}$  The number of users currently accessing the logging file.

DST The dst number of the logging data segment (LOGBUFF). (-1 = LOGBUFF not created yet)

The status of the logging process.

INITIALIZING = -1

INACT = 0

ACT = 1 RECOVERING = 2

 $\ensuremath{\mathsf{LGRUTO}}$  True if the automatic changelog facility was enabled. (Not used – for future use).

LCTYPE

LGTMPE
The type of destination file of the logging process.
DISC = 0
TARE = 1
SDISC = 2
CTARE = 3

 $\ensuremath{\mathsf{LGDEV}}$  The logical device number of the disc logging file or the disc logging buffer.

PIN

G.∞0.∞ 17- 25

User Logging Table

The PCB number for the logging process (PIN \* PCBSIZE).

Flag indicating a CMRMGELDG is pending (if true). (Not used – for future use).

LGNEHRUTO True if the autonatic changelog facility was requested for the new log file. (Not used – for future use).

GNEUTYPE

If a switch is pending, this will be the type of the new log process. (-1 = no switch pending). (Not used - for future use).

Sector number of the current extent in the disc logging file or the disc buffer file. (Disc buffer file has only 1 extent)

The number of records in the current extent (for disc logging) or the number available in the disc logging buffer.

NEXT R table relative pointer to the next entry in the logging table. (-1 = this is last entry)

At table relative pointer to the previous entry in the logging table. (-1 = this is first entry)

G.00.00 17- 26

User Logging Buffer

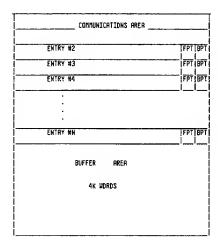
## User Logging Buffer

There  $\mu$ ill be one of these tables around for the life of any active user logging process. The table consists of three parts:

COMMUNICATIONS RATER – Information about status of the process, etc. that is common to all users of the process. Also the cells for messages to/from the process.

USER ENTRIES - Information for a specific user of the process. One of these for every user of a process (Setup by OPENLOG, released by CLOSELOG).

BUFFER RREA — Buffer used to hold logging records from all users before writing to the log file.



User Logging Buffer

<b>#</b> 0	COMMUNICATIONS AREA	z o
	- LOGGING -	
	- IDENTIFIER -	
4	SHITCH FLAG	4
5	NEW RUTO   NEW TYPE	5
6	AUTO TYPE	6
7	BUFFER DST	7
В	LOG PIN	10
9	NUMBER OF USERS	11
10	MAX NUMBER OF USERS	12
11	NEXT USER NUMBER	13
12	SLEEP COUNT	14
13	STRTE	15
14	MSG	16
15	LOG MSG	17
16	USEA MSG	20
17	LOG EARDR	21
1B	LOG DEVICE	22
19	BUFFER SPRCE	23
20	USED SPACE IN BUFFER	24
21	FILE SET NUMBER	25
22	LOG	26
	- ADDRESS -	
24	INPUT	30
	- RECORO -	
26	FILE	32
		00

	2 2 ZE	!
28	FILE	34
	SPRCE -	
30	TOTAL	36
	- RECORDS -	
32	MAX	40
	- SIZE -	
34	LAST EXTENT	42
35	EXTENT	43
36	ļ <del></del>	44
	-        -	<u> </u>
	i_ RESDURCE _ I	 
		<b>!</b>
40	<u> </u>	50
		İ
	<b> -</b>	į
		į
	ļ- <b>-</b>	
	-	
	THE WORLD DOOR DOOR	
48	IN USE HEAD PTA	60

FREE HERO PTA

User Logging Buffer

LOGIO 8LOGBUFF(D) LOGBUFF(4) LOGBUFF(5). (0:8) LOGBUFF(5). (8:8) LOGBUFF(6). (8:8) LOGBUFF(6). (8:8) LOGBUFF(7) LOGBUFF(8) SWITCH NEHRUTO NEHTYPE RUTD LOGTYPE BOST LOGPIN LUGPIN HUMUSER MAXUSER' USERND SLPCT STATE LOGBUFF (9) LOGBUFF(1D) LOGBUFF(11) LOGBUFF(12) LOGBUFF(13) HSG LOGBUFF (14) LOGBUFF (14) LOGBUFF (15) LOGBUFF (16) LOGBUFF (17) LOGBUFF (18) LOGBUFF (19) LOGMSG USERNSG LOGERR LOGDEV BSPACE BUFUSEO VSETND LOGBUFF(20) LOGBUFF(21) LOGADOR INBUFREC FSIZE FSPACE' DLOGBUFF(11) DLOGBUFF(12) DLOGBUFF(13) DLOGBUFF(14) TAECS MAXESPACE LASTEXT' LOGBUFF(34) LOGBUFF(35) RESDURCE -OLDGBUFF(18) LOGBUFF(48) LOGBUFF(49)

G.00.00 17- 29

61

6.00.00 17- 30

# Veer Logging Buffer

User Logging Buffer

LOGID
The name of the logging process.

49

SMITCH'
True if log file suitch is pending. (Not used - for future use).

NEURUTO
True if the automatic changelog option hae been specified for the new log
file. (Not used - for future use).

NEWIYPE If a suitch was requested, this will be the type of the new logging file. (-1  $\circ$  no suitch pending) (Not used - for future use).

 $\ensuremath{\mathsf{RUTD}}$  True if the automatic changelog option was specified for the current log file. (Not used – for future use).

LDGTYPE

The type of destination file for the logging proceed.

OISC = 0

TAPE = 1

SOISC = 2

CTAPE = 3

BOST The data eegment number of this table.

LOGPIN
This is the PCB number for the logging process (PIN\*PCBSIZE).

 $\ensuremath{\mathsf{NUMUSER}}$  The number of users currently accessing the logging file.

MAXUSER'

The maximum number of users allowed to access the logging file.

The next sequential number to be assigned users accessing the system. It uill get incremented for every unique DPENLOG - used as the  $\log$  # in the logging record format.

SLPCT The number of users currently waiting for activation by the logging process.

The state of the user logging process.

INRCTIVE = 0

RCTIVE = 1

MSS.

An internal message word used to indicate an error or operator request.

6 - Continue processing, all is fine.

2 - Suspend - error reading buffer file or writing to serial file

3 - Stop - set when issue :100 logid,STDP or when an EOF condition is found on the disc log file.

Ueer Logging Buffer

LOGMSG

nessages from the logging procese.

6 - Continue processing, all is fine.
15 - EOF - if there are no nore extents available to be

allocated.

12 - Disc space - could not allocate the new extent because no epace left in the group.

9 - Write error - error occurred while writing to log file

outshood

A meseages from the user process.

6 - Continue processing, all is fine.

12 - Olisc space - user process needs another extent allocated for disc logging.

LOGERR
Last error found. After changelog:
+N - File System error mumber encountered
D - No error
-1 - New disc log file was not empty
-2 - New disc log file did not have file code LOG
-3 - New disc file is too small
(Not used - for future use).

The logical device number of the current extent of the disc log file or the disc buffer file (buffer file has only 1 extent).

BOPHLE
The amount of space, in records, that are currently available to the users. On the last block of the last extent, one record uill be saved by the logging process so that the proper close information can be posted to the file either the trailer record (if the log logging process is stopped) or the change'to'new record because of an EOF condition (and the RUTD option had been specified).

BUFUSEO ourosed.

The number of records currently in the buffer. On all extents, except the last extent SUFSPACE-BUFUSED = 32 (number of records in a complete block). However, on the last block of the last extent this will NOT be true since one record is always held in reserve by the logging process.

VSETNO
This shows the order in the log file "set" of the currently opened log file.
(Not used - for future use).

The disc address of the current extent of the disc log file. If serial file, this is the disc address of the disc buffer for the file.

The record number of the next block to be written to the logging destination file or the disc logging buffer for serial files. (Used as an offset into the current extent for the writes - since each record is one sector in length).

#### User Logging Buffer

#### FSIZE

The current extent size of the logging destination file or disc logging buffer file for serial destination files. (on the last extent this will be the last extent size ninus 1).

The space in records that renains in the current extent of the disc logging destination file or disc buffer for tape destination files. (On the last extent of the disc log file, this is the amount of space minus 1).

The total number of records written to the logging destination file (including those records currently in the buffer).

The total file size, in records, minus 1. (Need that last record to post close information).

LASTEXT' The extent number of the final extent in the disc logging file or disc buffer file.

#### EXTENT

The current extent number of the disc logging file or disc logging buffer.

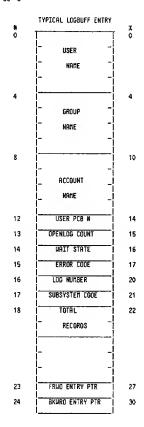
NESOURCE + 0 = Owner PCB number
RESOURCE + 0 = Owner PCB number
RESOURCE + 1 = Nead of inpeded queue PCB number
RESOURCE + 2 = Tail of inpeded queue PCB number
RESOURCE + 3 = Queue length

WHERO R table relative pointer to the first entry into the logging data segment. (-1 = no entries currently in use)

A table relative pointer to the first free entry in the logging data segment. (-1 = no free entries)

G.00.00 17- 33

#### User Logging Buffer



G.00.00 17- 34

### User Logging Buffer

BINDEX	=	BYTE INDEX TO CURRENT ENTRY
INCEX	=	WORD INGEX TO CURRENT ENTRY
OINOEX	=	OGUBLE INCEX TO CURRENT ENTRY

USER Group Acct BINGEX BINGEX+8 BINGEX+16

UPIN OPENCNT USTATE INGEX+12 INGEX+13 INGEX+14 INGEX+15 ERROR

INOEX+16 INOEX+17 RECS = OINOEX+9

LGNUM

NENTRY PENTRY

USER The name of the user who opened the logging file through this entry.

 $\ensuremath{\mathsf{GROUP}}$  The group of the user who opened the logging file.

INOEX+23 INOEX+24

# ACCT The account of the user who opened the logging file.

UPIN The PCB number of the user process (PIN  $\star$  PCBSIZE).

### OPENONT

Counter of how many times this user called OPENLOG. (Incremented for every OPENLOG, decremented for every CLOSELOG). (Not used - for future use).

USTRIE
The wait status of the users process.
IMACTIVE = 0
ACTIVE = 1

ERROR
Used to hold error information for this user.
-1 = No room in disc (or disc buffer) and NOURIT.
0 = 0.K.

LGNUM
The logging number assigned to the user. (From USERNO in global area to be used as log # in the log record).

SCOOE The subsystem code for the caller. This applies only to privileged callers.

The number of records written by this user.

### User Logging Buffer

At table relative pointer to the next entry in the logging data segment. (-1 = this is the last entry)

A table relative pointer to the previous entry in the logging data segment. (-1 = this is the first entry)



#### User Logging Identifier Table

ENTRY SIZE = #33 words DST X41

Table containing an entry for each potential logging process. Entries are added via :GETLOG and released via :RELLOG.





#### ENTRIES

MENTRIES ENTRYSIZE

LIDTRB(1) LIDTRB(4)

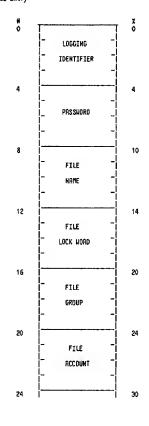
MENTRIES
The maximum number of entries in the table. (i.e. maximum number of user logging processes. 1 entry for every process – activated or not).

ENTRYSIZE
The size of each entry in the table.

G.00.00 17- 37

Logging Identifier Table

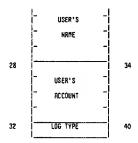
#### Typical Entry



G.00.00 17- 38

## Logging Identifier Table

# Typical Entry (Cont.)



#### BYTE ENTRIES

LIO PH FNAME' LH FGROUP FRECT UNAME URCCT BLIOTAB BLIOTAB(8) BLIOTAB(16) BLIOTAB(24) BLIOTAB(32) BLIOTAB(40) BLIOTAB(48) BLIOTAB(56)

## HORO ENTRIES

LIDTAB(32)

LIO The logging identifier name. This is a maximum of eight characters long.

 $\ensuremath{\text{PM}}$  The pass word for the logging identifier. This is a maximum of eight characters long.

The following is the fully qualified file name of the current log file.

FMAME'
The name of the destination file.

LW

The lock word on the destination file if the file is on disc.

FGROUP

# Logging Identifier Table

The group that the file resides in.

FACCT
The account that the destination file resides in.

UNRME The name of the user who created the logging identifier.

UACCT
The account of the user who created the logging identifier.

TYP
The status of the entry. -1 = null entry
0 = disc logging file
1 = tape logging file
2 = serial disc logging file
3 = cartridge tape logging file

#### User Logging Record Formats User Logging Record Formats Logging Record Format TARILER RECORD (STOP) AECDRD SIZE = 128 words USER AREA = 119 words 0 2 3 4 6 127 rec#|cksum|code |time|date| logid LOG RECORD AT DPENLOG **NULL RECORD** 2 3 4 6 7 11 12 24 25 127 2 127 3 4 6 7 rec#|cksun|code |time|date| logid|log# creator rec#|cksun|code |tine|date USER OR SUBSYSTEM/CONTINUATION LDG RECORD (from WRITELOG) BEGIN TRANSACTION MARKER 3 4 6 7 8 9 127 127 2 3 4 6 7 8 9 rec#lcksun|code |time|date|log#|len user area rec#|cksun|code |time|date|log#|len| user area LOG RECORD AT CLOSELOG END TRANSACTION MARXER 2 3 4 6 7 11 12 24 25 127 2 3 4 6 7 8 9 127 rec#|cksun|code |time|date |logid|log# creator pcb rec#|cksun|code |tine|date|log#|len| user area CARSN MARKER CODE DEFINITION 4 6 7 127 2 3 CDDE.(8:8) = rec#|cksun|code |time|date MEADER RECORD (START/RESTART) 2 3 4 6 11 127 rec#|cksun|code |time|date |logid G.00.00 17- 41 G.00.00 17- 42 User Logging Record Formats Measurement Information Table DATA FIELDS OF LOG RECDADS MERSINFOTAB DST = 59 (% 73)

#### NOTE:

- 1. The checksum algorithm uses the exclusive or (XDA) function against a base of negative one.  $\,$
- 2. Null record is used for filler.
- 3. The code word of the logging record can contain a subsystem code defined by the user in the first half of the word (0:8). User logging allows privileged users to pass this code in the index parameter of the <code>Openlog intrasic</code>.
- 4. The "len" field will contain the entire length of the data in the transaction (i.e. the length passed to WRITELDG, BEGINLOG, ENDLOG). If a continuation record is part of the transaction, it will also contain the entire length of the data. For example, a length of 440 was passed to the intrinsic. The "len" field of the first record will be 140, the "len" field of its continuation record will also be 140 even though the actual amount of data found in the first record will be 119 and the data found in the continuation record will be 21. (Positive length = # words, negative length = # bytes)

		_
		MERSLOEV
		Measplab
		MEASOSTN
Reserved	3	
for MERSI control	4	-
		• !
	6 I	<u>:</u>
	7 I	
	11   	
Reserved	12   	<u> </u>
for	13	l
performan tuning	14	
paranete	15 I	ŀ
	16	·
	17	- 
		MERSSTATX-
		MEASPADC- XDSBANX
		MERSPADC- XDSBASE
		MEASPROC- XDSNUM
	24   CLASS 14 STATISTICS XDS BANX	ĺ
	25   CLASS 14 STATISTICS XDS BASE	·
		•

## Measurement Information Table

26	Ī	CLASS	14	STRI	IST	ICS	xos	NUM.	<u>-</u>
27	1	CLASS	13	STAT	IST	ICS	xos	BANK	i
30	1	CLRSS	13	STAT	IST	ICS	XDS	BASE	
31	1	CLASS	13	STRI	IST	ICS	XOS	NUM.	
32	I	CLASS	12	STAT	IST	ICS	KDS	BANK	Ī
33	1	CLASS	12	STAT	IST	ICS	XOS	ease	l
34	I	CLASS	12	STAT	IST	ICS	XOS	NUM.	I
35	Ī	CLASS	11	STAT	IST	ICS	xos	BANK	l
36	1	CLASS	11	STAT	IST	ICS	XDS	BASE	
37	I	CLASS	11	STAT	IST	ICS	XOS	NUM.	ì
40	1	CLASS	10	STAT	IST	ICS	xos	BANK	1
41	1	CLASS	10	STAT	IST	ICS	XOS	BASE	ı
42	I	CLASS	10	STAT	IST	ICS	XDS	NUM.	Ī
43	Ī	CLASS	09	STAT	IST	ics	XDS	BANK	İ
44	1	C LASS	09	STAT	IST	ICS	XDS	BASE	1
45	I	CLASS	09	STRT	IST	ICS	XDS	NUM.	

## Measurement Information Table

1		
reserved .		Ī
neasurement .		I
.	<u>I</u>	ı
50	ICLASS O ENABLEO   CLASS 1 ENABLEO   COUNT	Ī
51	CLASS 2 EN.CNT.   CLASS 3 EN.CNT.	I
52	CLASS 4 EN.CNT.   CLASS 5 EN.CNT.	Ī
53	CLASS 6 EN.CHT.   CLASS 7 EN.CHT.	Ī
54	CLASS 8 EN.CHT.   CLASS 9 EN.CHT.	Ī
55	CLASS 10 EN.CHT.   CLASS 11 EN.CHT.	Ī
56	CLASS 12 EN.CNT.   CLASS 13 EN.CNT.	Ī
57	CLASS 14 EN.CNT.   CLASS 15 EN.CNT.	Ī
į 60		Ī
61		I
reserved for 62 shared		I
clock 63		Ī
user 64		Ī
65	I	Ī
66		I
67		1
		-

6.00.00 17- 45 G.00.00 17- 46

## Measurement Information Table

		_							
	70	I	H	FI	LAG		ı	8	
shar	ed 71	Ī		X	SI			Ī	
cloc	k 72	Ī		K	)S2			Ī	
interfa	ce 73	Ī		00	OUNT			Ī	
cell	s 74	Ī		DI	LIMIT			Ī	
	75	Ī		T	COUNT			Ī	
	76	Ī		Ţ	LINIT			I	
	77	Ī		DI	LABEL			1	
	100	Ī	MONI 1	OR BUFFE	R INOEX			 I	SMONIDX
	101	Ī	MERS	BUFFER				Ī	MERSBUFO
	102	Ī	MERS	BUFFER 1		<b></b> -		Ī	MERSIOX
reserve		Ī	MERS	ENABLEO				Ī	MERSMSKO
even	t 104	Ī	MERS	ENABLED	FLAGS			ı	neasmsk1
loggin	105	Ī	MERS	BUFFER (	BANK			Ī	MERSBUFBANK
	106	Ī						Ī	
	١.	Ī						Ī	
	١.	Ī						ı	
	116	Ī						1	
	117	Ī							
	١	-							

- M: Interrupt has missed due to last interrupt handling.
- A: Current interrupt handling active.

#### Message Files

#### CHAPTER 18 MESSAGE FILES

### Message File Data Structures

This chapter contains the data structures necessary to support nessage files. The first section details the message file's version of the familiar file system data structure; ie, the file label, file control block, access control block, etc..

The second section shows the tables used by the basic IPC mechanism which is a set of internal, MPE procedures designed to support the "boundary conditions" of IPC files. For example, signaling a no wait reader that its record has arrived. See the section's introduction for a detailed description.

#### File Label/FC8 Extent Map

	End of file block	Start of file block
Disc addr of extent 0		•
Disc addr of extent 1	Ü	•
	-	•
Disc addr of extent 2		•
		•
Disc addr of extent 3		•
Į		•
1	l .	•
		•
Disc addr of extent n-1		٧
		-
Disc addr of extent n		
	l	

The EOF and SDF are examples only, neant to show:

1) The start of file moves into the extent map as records are read 2) The file can wrap around and, hence, cause the SDF to be greater than the EDF.

When a file becomes empty the SOF and EOF are reset to the first block of extent zero.

Each extent is composed of a number of blocks. Extents all have the same number of blocks. Extent zero also contains space for the file label and user labels in the exact same format as standard files. Starting with block zero, sufficient blocks are allocated to the file label/user labels to satisfy their space requirements.

Extents outside of the SDF/EOF range may not exist. They are deleted at close time when there are no more writers accessing the file.

G.00.00 18- 1

#### Message Files

#### 8lock Structure

**********			
Eurot asua format as atomical			
Exact same format as standard   variable length blocks. 			
**********			

Separating the data portion of the records from their header enables the standard file system access procedures to read the records with no knowledge that they are  ${\tt msg}$  file records.

#### Record Format

Number of bytes in record
Fi 4-4
First data word of record
` 
Last data word of record

Length word's value does not include itself.

G.00.00 18- 2

#### Message Files

## Header Format

CITCI	ı	Neader		0
Writer's ID				-1

C (0:1) - Set on if this was the last record written before the system crashed. This bit is set on by the first open on the file after the crash.

LC (1:1)- Valid only for close headers. Set to one if this is the last writer to close the file.

Type(8:8)- O data 1 open 2 close

Message Access Control Block

Notes:
1. Words/fields that do not pertain to message files are left

This diagram shows the "combined" RCB as it appears to the message access procedures (the procedures in IPC). Thus it is a combination of the LRCB and the PRCB.

-5	DST number of the PACB	-5	
-4	PRCB control block vector table address	-4	
-3	DST number of the LACB	-3	
-2	<del></del>	-2	
-1			
0	Size of the RCB including buffers (words)	0	
1	File Number	1	*
2	File name	2	×
,	\ \	\	*
6	Foptions	6	*
7	Aoptions	7	×
8	Record size (bytes)	10	*
		1	

#### Message Files

9	Block size (words)	11 *
10		12
11	Carriage control code (uriters)	13 *
12	No wait I/O target	14 *
13	No иаıt I/D count	15
14	Error code	16 *
15	Transmission log (units same as last read/write)	17 *
16	Total number of unread records (includes opens	20
17	and closes)	21
18	Block number of the file's tail (relative to the	22
19	start of file block)	23
20	Logical record transfer count	24
21		25
22	Physical block transfer count	26
23		27
24	DST REL ADDR of Read Header	30
25	DST REL ADDR of Write header	31
26	FC8 DST	32
27	FCB vector table offset	33
28	Share count ( number of LACBs )	34
29	Access class, status, etc.	35
30	Logical device number	36
31	lirt buf indx    # buf - 1	37
32	DST relative address of next read record	40
33	Size of the buffer (words)	41
34	Spare	42
<b>3</b> 5	FMAVT index	43
36	Number of read LACBs	44

#### 37 | Type and disposition Access nask | Records per block |O|# rd buf | # ut buf |er |qu |n |c |d |s |f 39 47 Misc. msg file flags Number of free word in the current free record 51 Number of free records 42 52 53 43 Number of mondata records in the file 45 55 46 56 47 #open records | # read requests last read error | last write error 48 60 49 OST relative address of the next write record 50 Spare 63 51 Spare 52 DST rel address of the PACB DST rel address of the LACB 53 65 54 DST relative address of the stack RCB Stack DST relative address of DB 55 Target area's OST number 56 57 Reserved for calling parameters 58 72 59 73 60 Reserved for the stack marker from file system 74 intrinsics 61 64 User's soft interrupt plabel 100\* 65 | Number of seconds to wait on boundary condition | 101\*

Message Files

66	D[Ex Nd Vr Bt Cls  C   Carriage control	102*
67	Reply Port (basic IPC port)	103*
68	Writer ID	104*
69	Control block index for nowalt writer record buf	105*
70	OST relative addr of nowalt writer record buffer	106*
71		107*
72	No wait I/D resultant error code	110*
73	No wait I/D resultant transmission log	111
74	urite wait queue (basic IPC port)	112
75	Read wait queue (basic IPC port)	113
76	Length of record in bytes	114
77	Head record's record type (same values as header)	115
78	Head record's uriter IO	116
79	Misc. flags   Record type	117
80	Size of record + count + header words	120
81	Completor ID   Waiter ID	121
82	Local flags	122
83	Target DST nunber	123
84	OST relative address of target area	124
85	Length of target area	125
86	Waiter's reply port, O if using RCB compltn area	126
87	Waiting process's PIN	127
88	Waiting process's pin	130
89	Waiter's soft interrupt plabel	131
90	Resultant error code	132
91	Resultant transmission log	133
92	OST rel address of first buffer	134

G.00.00 18- 6

Message Files

Message Files

			buffer		- 1
ţ	 	 <b>-</b>		 <b>-</b>	 

\* Value is private to a particular accessor.

Word Field Description

6.00.00 18- 5

Rccessor's local flags.

(0:1) 0 1 - have not yet issued an FREAD/FURITE against the file.

(1:1) ex 1 - extended wait node.

(2:1) nd 1 - do not destroy the next record read.

(3:1) vr 1 - uriter has not yet uritten his first record (ie., he is a virgin).

(4:1) bt 0 - transmission log should be expressed in Hords.

1 - "" bytes.

(5:1) cls - Not currently used (reserved for group IPC standard).

(6:1) C - No wait completion message is in LACB area.

(8:8) car ctl- carriage control character to be used for the writer's record (a value of one indicates no carriage control character).

Message Files

Word Field Description File's global flags.

(1:4) - number of read buffers (5:4) - number of write buffers (9:1) er 1 - extended read (10:1) qu 1 - one or nore writers has been queued on the

wait queue. (11:1) n 1 - wait msg is located in the RCB

(12:1) c 1 - completion msg is located in the ACB

(13:1) d 1 - the current write buffer has dirty bit set

(14:1) s 1 - the start of file is block zero

(15:1) f 0 - the RCB buffers have not been filled

#### Message Files

#### MMSTAT Definitions

Dotal Value	Event Type	Parameter 1	Parameter 2
72/0	Read init	# free rec	
72/1	Read compl	(0:8) error, (8:8) ID	Number of records
72/2	Urite init	(0:8) # rec, (8:8) ID	Number of free records
72/3	₩rite compl	(D:8) error, (8:8) ID	Number of free records
72/4	Control	(D:8) error, (8:8) ID	(0:4) func, (4:12) parm
72/5	EDF	(D:8) error, (8:8) ID	Number of records
72/6	Dpen	(0:8) error, (8:8) ID	Number of records
72/7	Close	(8:8) #free, (8:8) ID	Number of records
72/10	Initiation	D	(0:8) fix, (8:8) update
73/0	Put record	(0:8) error, (8:8) ID	(D:3) rec type, (3:13) number of records
73/1	Delete rec	(0:8) error, (8:8) ID	(0:3) rec type (3:13) number of records
73/2	Delete blk	Start of file block #	End of file block #

#### Notes:

- 1. The aa/bb notation in the "octal value" column denotes type/subtype. Type is the actual MNSTRT event number. Subtype is (D|4) of parameter 0.
- Several items can possibly exceed their fields, in that case the bits beyond the field are lost. These items are number of records, number of free records, start of file, and end of file.

#### Message Files

3. Parameter word zero has a common format for all the MMSIRT events.

Field	Description Event's subtype.		
(0:4)			
(4:2)	File's state  0 - enpty 1 - partually full 2 - only a fraction of a free record is left 3 - completely full		
(6:1)	Nonzero indicates that there is one or more waiting readers.		
(7:1)	Monzero indicates that there is one or more waiting writers.		
(11:1)	Nonzero indicates that the write has a carriage control character.		
(12:4)	Flags local to the accessor.  (12:1) - the accessor has done no FREADs/FURITES  (13:1) - extended uait  (14:1) - nondestructive read  (15:1) - writer has not written any records		

6.00.00 18- 9

Message Files

#### File System Basic IPC Definitions

The objective of this set of uncallable procedures is to provide a simple ipc nechanism to support the ipc file access procedures. It enables one process to send short, control messages to another process.

### General Behavior

#### FCPDRTDPEN Procedure

The heart of this mechanism is the port. A process desiring to receive messages would first open (create) a port. This process is termed the "port manager." When the port is created, a port number is returned to the opener. Since the port number value cannot be known in advance, potential senders need some method of obtaining the port number from the port manager.

Both the ports and the messages are contained in a single disc resident data segment. There can be a total of over thiry-five hundred open ports and outstanding messages Thus neither ports nor message blocks are scarce resources.

## FCPORTSEND Procedure

This procedure sends a D to 5 word message to a port. Optionally a timeout value may be specified which will limit the duration the message will remain attached to the port. Expiration of the timeout causes the message to be deleted from the target port's queue and placed on the sender's reply port (specified by the sender in the FCPDRISEND procedure call).

#### {FCPDRTRECEIVE}

Reads and deletes the head message from a port. The sender's return port number is also given to the receiver, enabling him to send a reply message.

#### {FCPDRTCLDSE}

Demolishes the port.

#### {IPC file's use of this mechanism}

All open message files have two ports open for the file (read wait queue and write wait queue), plus one port per accessor (reply port). Their use is described in the following.

6.00.00 18- 10

### Message Files

Reader and writer wait queues) R When an empty message file is accessed by more than one reader (share), then there must be a way of having the readers' FRENDs satisfied in the same order that they were issued. That is, there must be queue of waiting readers. The ipc access procedures accomplish this by dedicating a basic ipc port as a "read wait queue." Whenever a reader's request is stalled because the file is entry, a message is sent to the read wait queue. Subsequent FRENDs by other processes will queue up behind the first reader in a FIFD manner. An FIRKITE will take the first entry from the wait queue and send a "read may be done" message to the reader's reply port.

In a like manner multiple writers will queue on the write wait queue when the file is full.

{Completion notification for nowait I/D}

The IDWAIT intrinsic maits for a message to be sent to the reply port (s) of the specified user files.

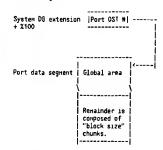
#### {Timeouts}

When an accessor encounters a boundary condition (ex, a reader accesses an empty file), it may specify that the condition nust be satisfied in x seconds (FEDNIROL 4). To this end the 1pc access procedures merely issue the FCDPRISEND to the wait queue with the user's timeout value specified. The timeout will tear the message from the wait queue and place it on the accessor's reply port.

# Message Files

# Port Oata Structures

### Port Data Segment



The chunks are a combination of free entries, ports, message queue entries, and timer list entries.

### Port With Two Dutstanding Messages

	-			-			
1	>	1	1-	>			
Port	ĺ	I TIGE 1	i	i	301	2 Ì	
1	1	1	İ	ĺ		i	
	-						

G.00.00 18- 13

Message Files

Port Number

Port index | Index into the port DST number array

Port DST Number Array

Located in System DB Extension Area.

64 | Port data segment number 64 65 | Reserved for a second port segment

G.00.00 18- 14

# Message Files

# Port Data Segment Global Area

0	Oata segment number of this port data segment	0
1	Block size in words	1
2	Total number of blocks	2
3	Maximum number of blocks	3
4	Current number of free blocks	4
5	Number of open ports	5
6	Head of free list	6
7	Tail of free list	7
10	Head of impeded process list	8
11	Tail of impeded process list	9
12	Head of timeout thread (TQE address)	10
13	TRLX of timeout	11
14	Value returned by TIMER intrinsic when	12
15	Timeout was initiated.	13
16	Head of port list (in units of port numbers).	14
17	Not used.	15

# Message Files

### Port

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
D   Head TQE address	D
1   Tail MQE address	1
2  E   N   Next port number in port list thread!	2
3  I  Subtype Port Pin number	3
4   Soft interrupt parameter one	4
5   Humber of MQEs in the port's queue	5
6   Number of sends to this port	6
7   Soft interrupt plabel	7
8   PIN of port's owner	10
0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15	
E Enable wake up bit 0 - Do not awaken the process 1 - Awaken the process	

 $\ensuremath{\mathsf{H}}$  type  $\ensuremath{\mathsf{R}}$  Action to be taken on an enabled port when a message is received.

0 - Ямаken the process on a message mait bit.

1 - Generate user software interrupt

2 - Generate system software interrupt

I Interrupt mode.

Subtype Soft interrupt subtype

# Message Queue Entry (MQE)

0 1 2 3 4 5 6 7 8 9 10 11 12 13	
O   Hext MQE entry; if last, (port addr) LOR	7   0
1   Port number of return port	1
2  Time List Entry (TLE),0=mo timeout,-1=time	d out  2
3   Parameter zero	3
4   Parameter one	4
5   Parameter tup	5
6   Parameter three	16
7   Parameter four	7

|0 |1 |2 |3 |4 |5 |6 |7 |8 |9 |10|11|12|13|14|15|

Timer entry definitions - 0 - no timeout 1 - timeout expired 2 - TLE address for a pending timeout

# File System Message Files

### Wait Message

parm#
0 - URITER ID
1 - LOCAL FIROS (differ with each accessor)
(0:1) - accessor just opened file
(1:1) - will wait on boundary condition if no symbiotic process
(3:1) - uriter has not written a record
(4:1) - transmission log in bytes
(8:1) - carriage control code
2 - DSI# of data buffer
3 - Address of data buffer (DST relative)
4 - Length of data buffer in bytes

# Completion Message

0 - Resultant error code 1 - Resultant transmission log in bytes

# Timer List Entry (TLE)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
0		0
1	Preceding TLE entry (0 if first entry)	1
2	Humber of milliseconds the timeout value	2
3	of this TLE is beyond the previous TLE.	3
4	Address of the affected MQE	4
5	Address of the MQE's port	5
6	Value of TIMER when this timeout expires	5
7	(Milliseconds)	7
	0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15	

# MMSTRT Oefinitions

Octal Value	Event Type	Parameter O	Parameter 1	Parameter 2
62	Open	Port number	Port DST num	Flags parameter
63	Receive completion	Port number	MQE address 15:1 Waitspc	Return port
64	Send	Port number	MQE address 15:1 Q type	Return port
65	Change status	Port number	0 = enable 1 = disable	Head MQE address
66	Abort	Port number	Parameter zero	Return port
67	Close	Port number	Port BST	# open ports left
70	Expand	Port OST num	# expand blks	Total # blocks
71	Timeout expired	Port num	MQE address	Return port

G.00.00 18- 18

G.00.00 18- 17

### Memory Resident Message Facility

### CHAPTER 19 HPE MEMORY RESIDENT MESSAGE FACILITY

### Overview of Facility

The nenory resident nessage facility of MPE V addresses the need for an efficient, simple, and uniform nethod for system code to send short status-type nessages to processes.

Each process is created with a "port" in the message harbor table (DST 271) which supports a set of message subqueues which are private to that process. There is a maximum of four subqueues per port in the initial implementation. This limit can be easily extended when new subqueues are required.

Any system code, even code running on the ICS, can send a nessage to any subqueue of any process. The destination process' PIN nust be known, any a priori conventions on subqueue mumber and nessage formats nust be established. The caller of SEMDNES nay optionally specify that the destination process be амакеnded from a message маіт.

Message can be any length up to the configured maximum. Message length is specified in the call to SENDNSG and RECEIVENSG. In the initial implementation, messages are limited to 4 mords in length. This maximum can easily be increased if the need arises.

By calling PORTSTRTUS, a process may at any time determine whether a specified subqueue is non-empty or obtain the subqueue number of the most urgent non-empty subqueue (lowest numbered one).

By calling RECEIVENSG, a process may receive the message at the head of the specified subqueue. This receive is optionally non-destructive.

 $\boldsymbol{\mathsf{A}}$  process can wait on a message wait, or on a combination of nessage wait and other wait types.

### Message Intrinsics

### SE NOM SG

Procedure SENONSG(Destpin, Subqueue, Msglength, Flags); Value Destpin, Subqueue, Msglength, Flags; Integer Destpin, Subqueue, Msglength; Logical Flags; Option Privileged, Uncallable;

Destpin, Subqueue, and MsgLength have to be within range or a System Failure 622 will occur.

The caller of SENDMSG stacks the message contents before calling the procedure. SENDMSG expects the first nsg word to be at G-7-MsgLength, and the last nsg word at G-8. The message contents at G-8 to G-7-MsgLength are deleted from the top of stack by the exit from SENDMSG to the caller.

Flags.(1:1) = 1 ==> Wake-up destination process from a message wait.

G.00.00

Menory Resident Message Facility

Return CC = CCG if process was already awake else CC = CCE.

### PORTSTRTUS

Logical Procedure PORTSTATUS(Subqueue); Value Subqueue; Integer Subqueue; Optiom Privileged, Uncallable;

When supplied a valid subqueue number, PORTSTATUS returns a true value if the subqueue is non-empty and a false value if the subqueue is empty.

When passed a -1 a subqueue parameter, PORISTATUS returns the subqueue number of the process' most urgent non-empty subqueue (the smaller the number, the more urgent the subqueue).

If all subqueues are empty, PORTSTRTUS returns CC - CCE. If at least one subqueue is nom-empty, PORTSTRTUS returns CC = CCG.

### RECEIVENSE

Procedure RECEIVEMSG(Subqueue, Msglength, Flags); Value Subqueue, Msglength, Flags; Integer Subqueue, Msglength; Logical Unicallable; Option Privileged, Uncallable;

Subqueue and Msglength has better be within range or a Systen Failure 622 will occur.

The caller of RECEIVENSG does an RSSEMBLE(ADDS Msglength) to make space for the nessage contents. RECEIVENSG stores the nessage contents into Q-8, Q-9,...,Q-7-Msglength. Q-7-Msglength contains the first word of the message.

Flags.(0:1) ==> do not release nessage from head of subqueue (nom destructive read).

Return CC = CCG if all subqueues were empty, else CC = CCE.

G.00.00 19- 2

# Memory Resident Message Facility

# Supporting Data Structures

### Message Harbor Table [DST #57 (%71)]

OST Index Number (%71)	
Data Segment Size	
Reserved	
Maximum number of PINS + 1	
Maximum Msg Size (6)	ĺ
Reserved	
Message Pool Head Pointer	į
Message Pool Tail Pointer	į
Rvailable Msg Frames Count	į
Head of impeded queue	į
Tail of impeded queue	į
Reserved 	
Ports (16 words each) (8 for header + 2 link words for each of 4 subqueues)	
Messages (6 words each)   (2 for header + 4 for data)	İ
	Oata Segment Size  Reserved  Maximum number of PINS + 1  Maximum Msg Size (6)  Reserved  Message Pool Head Pointer  Message Pool Tail Pointer  Rvallable Msg Frames Count  Head of impeded queue  Tail of impeded queue  Reserved  Ports (16 words each) (8 for header + 2 link words for each of 4 subqueues)  Messages (6 words each)

nasti	2TS F	vent:

# CHRPTER 20 HMSTRTS EVENTS

### MMSTATS Catalog Index

RLCSTBLK 20 024 (-) * FRERD RLLCCHEN 12 014 * FRERDL R BINKERD 233 351 (-) * FRERDLEREL BRERK 237 355 (-) * FRERDLEREL BRERK 237 355 (-) * FRERDLEREL CREDRITIO 142 216 * FRENDLEREL CREDRITIO 142 216 * FRENDLEREL CREDRITIO 142 216 * FRENDLEREL CLOSETRICEFILE 154 232 * FSETHODE CLOSETRICEFILE 154 232 * FSETHODE CLOSETRICEFILE 154 232 * FSETHODE CLONFICT 86 126 * FUNDLOCK COT RIT 86 126 * FUNDLOCK COT RIT 86 126 * FUNDLOCK COT RIT 86 126 * FUNDLOCK COT LIT 86 126 * FUNDLOCK COT LIT 86 126 * FUNDLOCK COT LIT 86 126 * FUNDLOCK COT LIT 86 126 * FUNDLOCK COT LIT 86 126 * FUNDLOCK COT LIT 86 126 * FUNDLOCK COT LIT 86 126 * FUNDLOCK COT LIT 86 126 * FUNDLOCK CONFIGTINFO 221 335 (-) * FURITIE CONFIGTINFO 222 336 (-) * FURITIE COPEL 13 357 (-) * FURITIE COPEL 13 357 (-) * FURITIE COPEL 155 233 * I/O CONPLETION 1 TINITIEST CRERD 147 223 * IOURIT CLINK REG CRERD1 147 240 * RINKEDC * RIPP DON CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 149 225 * PROCESS COMPLETE CUNDLOCK CURTITE 14	EVENT NAME	EVENT NO. DEC. %	EVENT NRME	EVENT NO. DEC. %
CREDRITIO 142 215 * REMARIE CRECIENDOV 14 016 * CCLOSE CLOSE 146 222 * FSEPTIODE CLOSTERREFEILE 152 230 * FUNLOCK COT RIT 86 126 * CONTROL COT RIT 86 126 * FUNLOCK COT RIT 86 126 * FUNLOCK COT RIT 86 126 * FUNLOCK COT RIT 86 126 * FUNLOCK COT RIT 86 126 * FUNLOCK COT RIT 86 126 * FUNLOCK COT RIT 86 126 * FUNLOCK COT RIT 86 126 * FUNLOCK COT RIT 86 126 * FUNLOCK COT RIT 86 126 * FUNLOCK CONTIG-INFO 221 335 (-) * FUNLTE CONTIG-INFO 223 337 (-) * FUNLTE COPENTRCEFILE 153 231 * FUNLTELIBRE COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRP COPENTRCEFILE 153 231 * IOBUTTRRPP COPENTRCEFILE 153 251 * INTITURE COPENTRCEFILE 153 251 * INTITURE COPENTRCEFILE 153 251 * INTITURE COPENTRCEFILE 153 251 * INTITURE COPENTRCEFILE 153 251 * INTITURE COPENTRCEFILE 153 251 * INTITURE COPENTRCEFILE 153 251 * INTITURE COPENTRCEFILE 153 251 * INTITURE COPENTRCEFILE 153 251 * INTITURE				
CREERD   142   216	RICSTALK	20 024 (-)	* FRERD	62 076 (-)
CREDRITIO		12 014	* FRERDDIR	64 100 (-)
CREBENTIO		233 351 (-)	* FRERDLABEL	76 114 (-)
CRBERTIO		237 355 (-)	* FRERDSEEK	68 104 (-)
CREDENTIO		139 213	*	
CRCHEHOV	CRADRITO	142 216	* FRENRME	80 120 (-)
CCLOSETRRCEFILE	CUDDELTO			,
CLIOSETRRCEFILE   154   232	CULOSE	146 222	* ESETMONE	72 110 (-)
COT RTT CGRBAGE 7 007	CCLOSE CCLOSETERCETLE	154 222	* ESDOCE	69 105 (-)
COT RTT CGRBAGE 7 007	CCONZEGG	154 232		79 117 (-)
COMPTIC-TINFO   221 335 (-)	CONTRUL	96 136		
COMPTIC-TINFO   221 335 (-)	COLEKII	00 120		66 102 (-)
Topentracefile	COHREHGE	7 007	* FUPURIE	63 077 (-)
Topentracefile	CONFIG-INFO	221 335 (-)	* LMKTIE	65 404 ( )
Topentracefile	CONFIG-INFO	222 336 (~)	* FMKTIEDIK	65 101 (-) 77 115 (-)
COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMPANY   COMP	CONFIG-INFD	223 337 (-)	* FURTIELHOEL	// 115 (-)
CDENTRICEFILE   153   231	COPEN	140 214	* PIPINIEKKUPI	192 300
TOPOLLIST				15 017
CRERD			* IOBUFTRRP	125 175
CRERD	CPOLLIST	155 233	* I/D COMPLETION	111 157 (-)
CRERD			* INITIRTE	84 124
CRERD1	CRERD	147 223	▼ TOMETI	67 103 (-)
CSDRIVER				
144   220	CRERD1	147 240		1 001
CSTORRIT				87 127 228 344 (-)
CSTORRIT	CSDRIVER	150 226	* MQNINIT	228 344 (-)
CURTITE	CSIDURIT	144 220	* MONDEF	229 345 (-)
DCIDCZRCK   231 347 (-)		149 225	* PROCESS COMPLET	TE 211 323 (-)
CERLLOCH		231 347 (-)	# Onucec	Λ ΛΛΛ
OISKBUGGRICHER		, ,	* QUE LDR	16 020
OISKBUGGRICHER	DERLLOCH	13 015	* QUIESCE	40 050
OISKBUGGRICHER	DERLESTRIK		* RELRESDURCES	23 027 (-)
OISKBUGGRICHER	Jeneco i Den	,	* REDCRCHE	90 132
No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.   No.	DESKRIGGREENER	200 310	* SEGTOINTT	
DISKERROR	DZONDOCCHICHEN	200 010		194 302
DISKERROR	OTSKRIIGERTEHER	201 311		194 302 195 303
SOUTT   DESTITE		100 144 (-)		6 006
DISKINTRPT 191 277 * SPECIRLRQ DQUE_LDR 17 021 *  * SPECRED  * STRRT I/O  * STRRTEGY	OTSVEVVOV	100 143 (-)	* SOFT OF TH	120 170
DISKINTRPT 191 277 * SPECIRLRQ DQUE_LDR 17 021 *  * SPECRED  * STRRT I/O  * STRRTEGY	OTENERROD	101 145 (-)	OO: . DE	120 170 236 354 (-)
* SPECRERD  * STRRT I/O  * STRRTEGY	DISKERNUN			2 002
* SPECRERD * STRRT I/O * STRRTEGY	DISKINIKAL			2 402
* STRRT I/O * STRRTEGY	DUUE_LDK	17 021		238 356 (-)
* STRRTEGY				193 301
SIRRIGUT			Dittiti Z) O	
		go (40 / )		83 123 8 010
	DISK TRAFFIC			
FCNECK 74 112 (-) * SYSPINS	FCNECK	74 112 (-)	* ZAZLINZ	224 340 (-)

### MMSTRTS Events

FCLOSE	81	121 (-)	*	SYSPINS	225	341 (-)
FCDNTRDL	71	107 (-)	×	SYSPINS	226	342 (-)
FETCHSEG	4	004	*	SYSPINS	227	343 (-)
FGETINFD	75	113 (-)	*	TERNLOGOFF	235	353 (-)
FIND DE	18	022	×			
FLOCK	78	116 (-)	*	TERMLOGON	234	352 (-)
FOPEN/(DR)	60	074 (-)	×	TERMRERD	230	346 (-)
FDPEN/(OR)	61	075 (-)	*	TERMURITE	232	350 (-)
FPDINT	70	106 (-)	×	UN MRP RG	88	130

G.00.00 20- 2

# MMSTRTS Events

### MMSTRT CRTRLOG INOEX

EVENT GROUP	DESCRIPTION DF GROUP	PAGE ND.
0	MEMORY MRNAGER	20-1
1	MEMORY MRNRGER/CRCNING	20-9
2	MEMORY MRNRGER	20-10
4	SCHEDULING	20-13
6	FILESYS	20-16
7	FILESYS	20-25
8	FILESYS/CRCHING	20-30
9	OISC I/D TRRNSFER/CRCHING	20-31
10	OISC ERRORS	20-32
11	SID	20-33
12	DISC SPACE	20-34
13	DISC CRCHING	20-51
14	CS/3000	20-36
15	CS/3000	20-40
16	CS/3000	20-43
19	DISC CONTROLLER INTRPT	20-44
20	PRIVATE VOLUMES	20-47
21	PROCESS CRERTION RND TERMINATION	20~48
22	MONITOR CONFIG INFORMATION	20-49
23	TERMINAL I/D	20-53

### MMSTRTS Events

# MMSTRT Event Group O (Memory Management Events)

_		
Ŀ,	rent	o

EVENT NRME: QONSEG
DESCRIPTION: ABSENCE TRRP ON CODE/DATE SEGMENT

CRILING MODULE: KERNELC CRILING PROCEDURE(S): QUEUEDNSEGMENT

PRRRMETER DESCRIPTION

# P1,P2 = Segment Identifier

P1.(0:4) = Segment type field 0 => Data Segment 1 => SL Segment 2 => Program Segment 3 => Cache Domain

P1.(4:12) = Program index into CSTBLK (type 2 only)

= Segment Number

P3 = SLL Pointer (SLL table relative)

P4 = STRTUS (in stack narker) of calling (trapping) segment

P5,P6 - Unused.

```
Event 1
```

EVENT NAME: MAKEOC DESCRIPTION: MAKE SEGMENT AN OVERLAY CRNDIDATE - RELEASE SEGMENT TO THE PODL OF AVAILABLE SPRCE

CRLLING MDDULE: KERNELC CALLING PROCEDURE: MRKEDC

# PRRRMETER DESCRIPTION

P1,P2 = Segment Identifier

P1.(D:4) = Segment type field
D => Data Segment
1 => SL Segment
2 => Program Segment
3 => Cache Domain

P1.(4:12) = Program index into CSTBLK (type 2 only)

= Segment Number

P3 = Bank of region P4 = Address of region

P5,P6 - Unused.

6.00.00 20- 5

MMSTRTS Events

### Event 4

EVENT NRME: FETCHSEG
DESCRIPTION: SEGMENT REQUEST (FDR I/D SYSTEM DR PROCESS)

CRLLING MDDULE: KERNELC CALLING PROCEDURE: FETCHSEGNENT

# PARRMETER DESCRIPTION

P1,P2 = Segment Identifier

P2 = Segment Number

P3 = Requester ID
.(0:1) = 1 => I/D System request
.(1:15) = Ldev #
.(0:1) = 0 => Process request
.(1:15) = Pin # of requesting process

.(1:1) = 1 => IDFREEZE REQUEST .(2:1) = 1 => BLOCKED LOCK REQUEST .(3:1) = 1 => LDCK REQUEST .(4:1) = 1 => FREEZE REDUEST

P4= .(13:3)= 0 => Segment already present = 1 => Segment is Recover Overlay Candidate = 2 => Segment already on its way in for someone (Segment In Motion In) = 3 => Segment not present -- must fetch (full fetch)

P5,P6 - Unused.

MASTATS Events

### Event 2

EVENT NAME: SPECIALRQ
DESCRIPTION: REQUEST OF SEGMENT EXPRNSION/CONTRACTION, UNLOCK,
UNFREEZE, IDUNFREEZE, LOCK, IOFREEZE, FREEZE

CRLLING MODULE: KERNELC, KERNELD, ININ
CRLLING PROCEDURES: UNLOCKSEG', IOFREEZE', FETCHSEGHENT-(KERNELC)
DLSIZE, ZSIZE, GETPMSEG, ALTOSEGSIZE,
RLTPWFILESIZE
STRCKOVERFLON
- (KERNELD)
- (THIN)

### PRRAMETER DESCRIPTION

P1,P2 = Segment Identifier

P1.(0:4) = Segment type Field
0 => Data Segment
1 => SL Segment
2 => Program Segment
3 => Cache Domain

P1.(4:12) = Program index into CSTBLK (type 2 only)

= Segment Number

P4 = For types (P3. (12:4)) = 0,2,3,5 => P4.(8:8) = LOCK DR IDFREEZE COUNT = 1,4 => P4.(0:8) = FREEZE COUNT = 6-15 => REQUESTED SIZE DF RRER IN HORDS

P5,P6 - Unused.

6.00.00 20- 6

# MISTATS Events

# Event 5

EVENT NAME: SEGID
DESCRIPTION: MEMDRY MANAGEMENT READ/WRITE OF SEGMENT FROM/TO
DISC QUEUED

CRILING MODULE: KERNELC CRILING PROCEDURES: PROCESSINITHSG, STRRTSEGURITE

PRRRMETER DESCRIPTION P1,P2 = Segment Identifier

P1.(4:12) = Program index into CSTBLK (type 2 only)

= Segment Number

P3 = Disc Request Index - (DRQ Table relative)

P4 = .(0:1) = 1 => WRITE START = D => RERD START .(1:15)= Ldev #

P5, P6 - Unused.

9.00.00 20- 7

G.00.00

### MMSTRTS Events

EVENT NAME: SIODOME
DESCRIPTION: MEMORY MEMORY SEGMENT READ/WRITE FROM/TO DISC
COMPLETE

CALLING MODULE: KERNELC CALLING PROCEDURES: SEGREADCOMPLETOR, SEGNRITECOMPLETOR

### PARAMETER DESCRIPTION

### P1,P2 = Segment Identifier

P1.(0:4) = Segnent type field 0 => Uata Segnent 1 => SL Segnent 2 => Program Segnent 3 => Cache Domain

P1.(4:12) = Program index into CSTBLK (type 2 only)

P2 = Segment Humber

P3 = Disc Request Index (ORG Table relative)
P4 = .(0:1) = 1 => Write complete
= 0 => Read complete

P5,P6 - Unused.

### Event 7 (%7)

EVENT NAME: CGARBAGE EVENT DESCRIPTION: GARBAGE COLLECTION NRS JUST TAKEN PLACE

CALLING MODULE: KERNELC
CALLING PROCEDURE: COLLECTGARBAGE

### PARRMETER DESCRIPTION

P1 = BRNK OF SOURCE JUST MOVEO FROM
P2 = RODR OF SOURCE JUST MOVEO FROM
P3 = NOVEPROECAT, NUMBER OF PROES JUST MOVEO FROM
P4,P5,P6 - Unused.

MMSTATS Events

### Event 8 (210)

EVENT NRME: SURPIN DESCRIPTION: SURP IN A PROCESS

CRLLING MODULE: KERNELC CALLING PROCEOURE: SWAPIN

### PARAMETER DESCRIPTION

P1 = PIN OF PROCESS BEING SUAPPED IN
P2 = .(0:1) = 0 => BEING SUAP
= 1 => END SUAP
.(1:1) = 0 => NORIRE (PRETIEL SUAP DK)
= 1 => SUAPP REQUIRED
.(12:4) = 0 => PROCESS SUAPPIN COMPLETE
2 => NO RODM, HARD REQ HAY SUCCED
3 => NO RODM, HARD REQ FAILED
4 => SUAPPIN STOPPED - MORE URGENT ACTIVITY
8 => NO LOCK SPACE
P3 = HARDREQUEST = TRUE => NARYO REQUEST ON SUAPPIN
FALSE=> NORMEL

P4.P5.P6 - Unused.

6.00.00 20- 9

### MMSTRTS Events

### MMSTRT Event Group 1 (Memory Manager)

### Event 12 (%14)

EVENT MRME: RLLOCMEM
DESCRIPTION: FOUND A NOLE FOR A SEGMENT REPLACEMENT REQUEST

CRELING MODULE: KERNEEC
CALLING PROCEDURE: RESERVEREGION

# PARAMETER DESCRIPTION

P1 = REQUESTED SIZE IN PRGES
P2 = BRNK DF SELECTED REGION
P3 = ADDRESS OF SELECTED REGION
P4,P5,P6 - Unused.

# Event 13 (215)

EVENT NAME: DEALLDON DESCRIPTION: RELEASE REGION OF MEMORY TO AVAILABLE STATUS

CRLLING MODULE: KERNELC
CALLING PROCEDURE: RELEASEREGION

### PARAMETER DESCRIPTION

P1 = SIZE RELERSED IN PAGES
P2 = BANK DF RELEASED REGION BASE
P3 = ADDRESS OF RELEASED REGION BASE
P4,P5,P6 - Unused.

# MMSTATS Events

### Event 14 (%16)

Event Name: CRCHENGY
Description: R cache move (i.e. logical disc request) has
just completed.
Calling Module: CRCHESEG
Calling Procedure: ProcessCDTLogReqQue

G.00.00 20- 10

# Parameter Description

P1,P2 = Segment identifier of target DST (LDR'BUFOST)
P2.(0:1) = 1 then this is a stack.
P3 = Happed Donain CDT entry number
P4 = Transfer count
P5,P6 = Unused

### Event 15 (217)

Event Name: GET\_CDT
Description: Called when an entry in the CDT table is obtained or released.
Calling Module: CRCMESEG
Calling Procedures: Get'CDT'Entry, CDT'Free'Entry, CDT'Get'MD'Entry, CDT'Rel'MD'Entry

### Parameter Description

P1 = CDT entry number
P2 = Type of call
O = Free entry
1 = Get entry
2 = Get Mapped Domain entry
3 = Release Mapped Domain entry
P3 = If P2-3 then Ldev Entry number
P4,P5,P6 Not used.

### Event 16 (X20)

Event Name: QUE\_LDR
Description: Called when an LDR is queued onto the CDT
Calling Module: CRCHESEG
Calling Procedure: CDT'Queue'LDR

Parameter Description

= Mapped Donain CDT entry number = LDR entry index to be Queued = Queue type T12 - CDT inpeded queue X13 - CDT active queue P4.P5.P6 Not used.

### Event 17 (X21)

Event Name: DQUE\_LDR
Description: Called when an LDR is removed from the CDT queue.
Calling Module: CRCHESEG
Calling Procedure: CDT' Dequeue'LDR

Parameter Description

= Mapped Donain CDT entry number = LDR entry index being removed from the queue = Queue type %12 - CDT impeded queue %13 - CDT active queue P1 P2 P3 P4.P5.P6 Hot used.

### Event 18 (X22)

Event Hame: FIMD\_DE
Description: Called when need to find an assigned CDT
Device entry.
Calling Module: CROKESEG
Calling Procedure: COT'Find'DE

P1 = Ldev number of the CDT Device entry to be Found.
P2 = CDT Device entry
P3,P4,P5,P6 Not used.

G.00.00 20- 13

# MMSTAT Event Group 2

### Event -20 (-X24)

EVENT MANE: ALCSTBLK
DESCRIPTION: AEDUEST TO AESERVE A BLDCK OF ENTRIES IN THE CSTX

CALLING MODULE: KERNELD
CALLING PROCEDURE: ALCSTBLOCK

# PARRMETER DESCRIPTION

CST BLOCK INDEX RSSIGNED
DST RELATIVE INDEX OF HORD O
DF THE FIRST RESERVED CSTX ENTRY
NUMBER DF CSTX ENTRIES RESERVED P2=CSTX P4, P5, P6 - Unused.

### Event -21 (%25)

EVENT NRME: DEALCSTBLK DESCAIPTION: INDICATES THAT A CST EXTENSION BLOCK HRS BEEN DEALLOCATED

CALLING MODULE: KERNELD
CALLING PROCEDURE: DEALCSTBLOCK

PARRMETERS PARAMETER DESCRIPTION CST BLOCK INDEX ASSIGNED TO THE BLOCK OF CST ENTRIES OST AELATIVE INDEX OF UDRO O OF THE FIRST CST ENTRY TO BE RELEASED P1=EIX P2=CSTX

=(#ALLOCRTED CSTX ENTRIES-#ENTRIES BEING RELERSEO)\*4 P3=MCHT P4,P5,P6 - Unused.

G.00.00 20- 14

### MMSTRIS Events

### Event -23 (-%27)

EVENT NAME:RELRESDURCES
DESCRIPTION: RESDURCES (VDS.HAIN HEMDAY, ST ENTRY) RESERVED FOR THE
FOR THE SEGMENT NAVE BEEN RELEASED

CALLING MODULE: KERNELD

CALLING PROCEDURE: RELDRIASEG

PRRAMETERS PARRMETER DESCRIPTION

P1=NEW OB DST NUMBER P2=DELTA P AT EXCHANGEDB CALL

P3=STATUS RT EXCHANGEDB CALL P4,P5,P6 - Unused.

MMMSTAT Event Group 3

(NOT CURRENTLY ASSIGNED)

### MMSTRTS Events

### MMSTAT Event Group 4 (Scheduling)

# Event 40 (%50)

EVENT NAME: QUIESCE
DESCRIPTION: PROCESS SHITCH - STATE OF PROCESS SHYED

CALLING MODULE: KEANELC CRLLING PROCEDURE: DSP

# PARAMETER DESCRIPTION

```
P1 = PCBOO(CPCB)

.(0:1) = 1 => SRR - SCNEOULING ATTENTION REDUIRED

.(0:1) = 1 => CRIT - PROCESS IS CRITICRL

.(2:1) = 1 => MSIR - PROCESS IS CRITICRL

.(3:1) = 1 => MSIR - PROCESS HAS SIR

.(4:1) = 1 => PIOWR - PENDING PI, PROCESS CRITICAL

.(5:1) = 1 => NEWP - INCORE PROTECT CEMPIRED

.(6:1) = 1 => NEWP - INCORE PROTECT CEMPIRED

.(8:1) = 1 => NP - PRECENPT RECENPT

.(9:1) = 1 => LD - LONG UNIT

.(10:1) = 1 => SU - SHORT UNIT

.(11:1) = 1 => SU - SHORT UNIT

.(11:1) = 1 => SU - SHORT UNIT

.(12:1) = 1 => USEDD - USED A QUARTUM SITUE TRANSACTION

BEGAN

.(13:1) = 1 => NIPRI - NOLD INPERED PRIDRITY

.(14:1) = 1 => ALLOW SOFT INFERRUPTS EVEN THOUGH IN

SYSTEM CODE

.(15:1) = 1 => RITBK - PROCESS IN RIT BREAK
```

```
- MOURNING WAIT
- GLOBAL RIN WAIT
- LOCAL RIN WAIT
                                                                                                    - LOUNT KIN WHATT
- MAIT WAIT
- BLOCKED IO WAIT
- IO WAIT
- UCOP WAIT, RIT WAIT
- JUNK WAIT
- TIMER WRIT
                                                                                                  - TIMER WRIT
- INTERRUPT WAIT
- SON WRIT
- FORMER WRIT
- PROCESS WRITING TO UNINPEDED
- PROCESS WRITING FOR SIR
- PROCESS WRITING FOR TIME OUT
- PROCESS WRITING FOR MEMORY
```

### MMSTRTS Events

P3 = PCB13(CPCB) .(O:1) = 1 => DISPQ - PROCESS ON DISPRTCHING QUEUE

.(1:1) = 1 => L SCHEOULING CLASS .(2:1) = 1 => C SCHEOULING CLASS .(3:1) = 1 => C SCHEOULING CLASS .(4:1) = 1 => E SCHEOULING CLASS .(5:1) = 1 => E SCHEOULING CLASS .(5:1) = 1 => DIRER - PROCESS IS INTERRCTIVE .(6:1) = 1 => DORER - PROCESS IS COME-RESIDENT .(8:8) = PROCESS' SCHEOULING PRIORITY

P4, P5, P6 - Unused.

MMMSTAT Eyent Group 5

(SEE CHAPTER 18 FOR THESE EVENTS)

MMSTRIS Events

MASTAT Event Group 6 (FILESYS)

THESE EVENTS RRE FOR DEVELOPMENT USE DNLY AND ARE NOT WORMALLY ENABLED

Event -60(274)

EVENT NAME: FOPEN DESCRIPTION: OLD FILE OPEN CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPENDA

PARAMETERS PARAMETER DESCRIPTION

P1= FILE # (0:2)=2 -> NON-SPOOLER RECESS (0:2).NE.2 ->

P2= AOPTIONS SEE INTRINSICS MANUAL
P3= FILE LHBEL FORTIONS SEE INTRINSICS MANUAL
P4= RECORD SIZE
P5= FILE LHBEL BLOCK SIZE
P6= M OF BUFFERS

G.00.00 20- 17

MMSTRTS Events

Event -61(275)

EVENT MRNE: FOPEN'
DESCRIPTION: DLD FILE OPEN (CONTINUATION OF EVENT -60)

CALLING MODULE: FILERCO

CALLING PROCEDURE: FORENDA

PARAMETERS PRERMETER DESCRIPTION

P1= FILE LABEL FILE LIMIT nsu

P2= FILE LABEL FILE LIMIT

P3= FILE LABEL W OF EXTENTS

P4-P6 unused

Event -60(274)

EVENT NAME: FOREN
DESCRIPTION: NEW DISC FILE OPEN CALLING MODULE: FILEACC

CRLLING PROCEDURE: FOPEN

PARAMETER DESCRIPTION PARAMETERS

P1= FILE N (0:2)=2 -> NON-SPOOLER ACCESS (0:2).NE.2 ->
P2= ROPTIONS SEE INTRINSICS MANUAL

P3= FOPTIONS SEE INTRINSICS MANUAL

P4= RECORD SIZE

PS= BLOCK SIZE

P6= # OF BUFFERS

G.00.00 20- 18

MMSTRTS Events

Event -61(275)

EVENT NAME: FOPEN' OESCRIPTION: NEW DISC FILE OPEN (CONTINUATION OF EVENT -60)

CRELING MODULE: FILERCE

CALLING PROCEOURE: FOPEN

PRRAMETERS PARAMETER DESCRIPTION

P1= FCB FILE LIMIT

P2= FCB MAX W EXTENTS

P3= (0:8)= INITIRL RELOCATION EXTENTS

P4-P6 unused

9.00.00

# MISTRIS Events

# Event -62(176)

EVENT NAME: FREAD DESCRIPTION:

CALLING MODULE: FILEIO

CRELING PROCEDURE: FREND

PRARRETERS PRRAMETER DESCRIPTION

P1= FILE #

P2= RCBTLOG

TRANSFER COUNT

P3= FLAGS

(15:1) Buffer hit flag

Event -63(277)

EVENT MAME: FURITE DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FWAITE

PARAME TERS

PARAMETER DESCRIPTION

P1= FILE #

P2= TCOUNT

SEE INTRINSIC MANUAL

P3= FLRGS

(15:1) Buffer hit flag

G.00.00 20- 21

6.30 00 30- 22

## MASTRIS Events

# Event -65(X101)

EVENT NAME: FURITEDIR DESCRIPTION:

CALLING MODULE: FILEIO

CAULING MODULE: FURITEDIR

PARAMETERS. PARRMETER DESCRIPTION

P1: FILENUM

PZ= TCOUNT See Intrinsic manual

LSU

P3= FLAGS

(15:1) Buffer hit flag

P4= REC # ทรม

PS= REC #

P6= HOT USED

MMSTATS Events

MMSTATS Events

Event -64(X100)

EVENT NAME: FREADDIA DESCRIPTION:

CALLING MODULE: FILEIG

CALLING PROCEDURE: FREADDIR

PARAMETERS

P1= FILE #

P2= ACBTLOG

P3= FLAGS

P4= REC #

P5= REC #

P6= NOT USED

PARAMETER DESCRIPTION

nsu

LSM

TRANSFER COUNT

(15:1) Buffer hit flag

Event -66(X102)

EVENT NAME: FUPDRIE DESCRIPTION:

CALLING MODULE: FILEID

CALLING PROCEDURE: FLPORTE

PERAMETERS PPRAMETER DESCRIPTION

Pf= FILE #

P2= TCOUNT See Intrinsic manual

P3= FLAGS (15:1) Buffer hit Flag

P4-P6 not used

Event -67(X103)

EVENT NAME: ICHAIT DESCRIPTION:

CALLING HODULE: FILEID

CRELING PROCEDURE: IDWRIT

PARAMETERS

PRRAMETER DESCRIPTION

P1= FILE #

P2= RCBTLCG TRANSFER COUNT

P3= FLRGS

(15:1) buffer hit flag

9.00.00 20-13

AMSTRES Events

Event -58(2104)

EVENT MRME: FRENDSEEK DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FREROSEEK

PARAMETERS PRRAMETER DESCRIPTION

P1= FILE #

P2= FLAGS

(15:1) buffer hit flag

P3= REC # MSM

P4= REC # LSM

PS-P6 not used

Event -69 (X105)

EVENT NAME: FSPACE DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FSPRCE

PARAMETERS

PARAMETER DESCRIPTION

P1= FILE #

P2= DISPLACEMENT SEE INTRINSIC MANUAL

P3-P6

not used

G.00.00 20- 25

MMSTATS Events

Event -72 (2110)

EVENT MRME: FSETMODE DESCRIPTION:

CALLING MODULE: FILETO

CRILING PROCEDURE: FSETMODE

PARAMETERS PARAMETER DESCRIPTION

P1= FILE #

P2= MODEFLAGS SEE INTRINSIC MANUAL

P3-P6

not used

Event -74 (2112)

EVENT NAME: FCHECK DESCRIPTION:

CALLING MODULE: FILEIG

CALLING PROCEDURE: FCHECK

PARAMETERS PARAMETER DESCRIPTION

P1= FILE #

SEE INTRINSIC MANUAL P2= ERACACODE

P3-P6

not used

SMSTRIS Events

MMSTRT Event Group 7 (FILESYS)

THESE EVENTS ARE FOR CEVELOPMENT USE DNLY RHO ARE HOT MORRALLY ENABLED

Event -70 (X106)

EVENT MAME: FPOINT DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FPOINT

PARRMETERS PARAMETER DESCRIPTION

P1= FILE #

P2= REC # MSN

P3= L\$W LSW

24-26 not used

Event -71 (2107)

EVENT NAME: FCCNTROL DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FCONTROL

PARAMETER DESCRIPTION PARAMETERS

P1= FILE #

P2= CODE See Intrinsics manual

93-96 not used

0.00.00 20- 26

MMSIRIS Events

Event -75 (X113)

EVENT NAME: FGETINFO DESCRIPTION:

CALLING MODULE: FILEIO CALLING PROCEDURE: FOETINFD

PARAMETER DESCRIPTION PARRMETERS

P1= FILE #

SEE INTRINSIC MANUAL P2= FOPTIONS

P3= ROPTIONS SEE INTRINSIC MANUAL

P4-P6 not used

Event -76 (2114)

EVENT NAME: FREROLABEL DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE:

PARAMETERS PARAMETER DESCRIPTION

P1= FILE #

SEE INTRINSIC MANUAL PZ= TCOUNT

P3-P6 unused

5.00 00

### MMSTRTS Events

# Event -77 (2115)

EVENT NAME: FURITELABEL DESCRIPTION:

CALLING MODULE: FILEIO

CRLLING PROCEOURE: FURTTELABEL

PARAMETERS PARAMETER DESCRIPTION

PI= FILE #

SEE INTRINSIC MANUAL P2= TCOUNT

P3-P6 unused

Event -78 (2116)

EVENT NAME: FLOCK DESCRIPTION:

CALLING MODULE: FILEID

CHLLING PROLEDURE: FLOCK

PARAMETERS PARAMETER DESCRIPTION

HT= FILE #

P2= LOCKCONO See Intrinsics manual

93= COND CODE " " " "

G.00.00 20- 29

MMSTRTS Events

MMSTAT Event Group 8

Event -80 (Z120)

EVENT MANE: FREMANE DESCRIPTION:

CALLING MODULE: FILERCO

CALLING PROCEDURE: FREHAME

PARAMETER DESCRIPTION PARAMETERS

P1= FILE #

P2-P6 unused

Event -81 (X121)

EVENT NRME: FCLOSE DESCRIPTION:

CALLING MODULE: FILERCO

CALLING PROCEDURE: FCLOSE

PARAMETERS PARRMETER DESCRIPTION

Pt= FTIF #

P2= DISP

See Intrineic manual

P3= SECCODE

P4-P6

unused

MMSTRTS Events

Event -79 (2117)

EVENT MAME: FUNLOCK DESCRIPTION:

CALLING MODULE: FILETO

CALLING PROCEDURE: FUNLDCK

PARAMETERS PARAMETER DESCRIPTION

P1= FILE #

P2-P6 unused

9.90.00 20- 30

MMSTRTS Evente

Event 83 (2123)

Event Name: STRRTEGY
Description: Called to determine the type of strategy used based on who the caller of CDT/RTTACHIO is.
Calling Module: CACHESC Calling Procedure: CDT/STGRTEGY

Parameter Description

P1 = COT Mapped Curain entry
P2 = LDR entry index
P3 = Strategy
O - Unknown caller
1 - Unknown from File System
2 - Spooler
3 - Directory
4-7 - Unknown
8 - Gennessage
9 - File System, Oulesce I/O
10 - File System, sequential, MOBUF
11 - File System, sequential, MOBUF
12 - File System, sequential, BUF
13 - File System, stemptial, BUF
14 - File System, stemptial, BUF
15 - File System, SRR
15 - File System, KSRR
15 - File System, KSRR

### MMSTRIS Evente

### Event 84 (X124)

Event Name: INITIATE
Description: Called when starting/completing logical disc
request.
Calling Module: CACMESEG
Calling Procedures: CD1'Initiator, CD1'Completor

# Parameter Description

P1 x COT Mapped Domain entry number
P2 = LDR entry index
P3 x type
O = Initiator
1 - Completor
P4,P5,P6 Hot used.

### Event 86 (X126)

Event Mane: CDT\_RTT Description: Called from COT'RTTACHIO. Calling Module: CRCHESEG Calling Procedure: CDT'Rttachio

### Parameter Description

P1 = Ldev P2 = Function P3 = Flags P4,P5 = Parm1, Parm2 P6 = Count

### Event 87 (X127)

Event Name: MRP\_DOM
Description: Called when need to "map" a disc domain.
Calling Nodule: CACHESEO
Calling Procedure: CDT'MRP'CRCHEO'DOMRIN

# Parameter Description

G.00.00 20- 33

### MMSTRTS Events

### Event 88 (X130)

Event Name: UN MRP\_RG
Description: Called when disc domain no longer napped. (i.e. both
the logical and physical 1/0 ie complete).
Calling Module: CACHESEG
Calling Procedure: CDT MRP/CRCHED'REGION

# Parameter Description

P1 = CDT Ldev entry number P2 = Region CDT entry number P3,P4,P5,P6 Not used.

### Event 89 (X131)

Event Name: LINK REG
Description: Called when a disc donain gets linked into the Inked list of donains for an idev.
Calling Module: CACKESEG
Calling Procedure: LINK'CRCHED'REGION,UNLINK'CACHED'REGION

# Parameter Description

P1 = Type
0 = Link
1 = Unlink
P2.P3 = R6ddress of region base
P4 = COT entry number found in the header
P5 = W of pages
P6 Not used.

G.00.00 20- 34

P1 = New CDT entry number P2 = Returned CDT entry P3,P4,P5,P6 Not used.

### MMSTATS Events

# MMSTRT Event Group 9 (Disc I/O Requests)

### Event 90 (X132)

Event Name: REGCRCHE
Description: Called to see if caching will accept this
1/0 request.
Calling Module: CRCHESEG
Calling Procedure: REGUEST CRCHE

# Parameter Description

P1 = LDR entry index P2,P3,P4,P5.P6 Not used.

# Event -98 (2142)

EVENT HAME: DISK TRAFFIC DESCRIPTION: DISC I/O REQUEST HAS BEEN QUEUEO

### CRITING MODULE: HARDRES

## CRLLING PROCEDURE: RTTACHIO

PARAMETERS PRRAMETER DESCRIPTION P1=CNT DATA TRANSFER COUNT: WORDS IF >O; BYTES IF <0 P2=FLRGS.(0:4) P3=FNCT =0 ==>RERD =1 ==>WRITE =2 ==>CLOSE FILE =4 ==>CLOSE FILE

### MMSTRTS Events

# MMSTAT Event Group 10

### Event 100 (X144)

EVENT NRME: DISK ERROR DESCRIPTION: RECORD DISC ERROR

# CRILING MODULE: IDFOISC1

CRLLING PROCEDURE: FHODVR

# PARAMETERS PARAMETER DESCRIPTION

P1=DIPT(DSTRT) HRROWRRE STRTUS P3=10QP(QLDEY), QLDEYN LOR STOCOUNT&LSL(8))
=LDEY/SIO PROGRAM COUNTER

# Event 101 (X145)

EVENT NAME: DISK ERROR DESCRIPTION: RECGRO DISC ERROR CALLING MODULE: IOMOISCO

# CALLING PROCEDURE: MHOOVE

PRRAMETERS PARAMETER DESCRIPTION

P1=OIPT(DSTAT) NARDWARE STATUS
P2=SO QNISC
P3=IODP(QLDEV).OLDEV# LOR STOCOUNT&LSL(8))
=LDEV/SIO PROGRAM COUNTER

G.00.00 20- 36

### MMSTRTS Events

MMSTRTS Evente

MMSTATS Events

### MMSTRT Event Group 11

### Event -110 (X156)

EVENT NAME: START I/O DESCRIPTION: DRIVER INITIATOR FOR SIO DEVICE HAS BEEN CALLED

CALLING MODULE: HARDRES

CALLING PROCEDURE: SIODA

PARRHETER DESCRIPTION PARAMETERS

P1=IGGPL(DSTAT) LOR IDOPL(DLDEV).LDEVN
=(D:S) PCB ENTRY N OF PADCESS MAKING REDUEST
(S:S) LOGICIAL DEVICE WIDNER OF DEVICE FOR I/O
P2=IGGP(DGDCT)=MORD COUNT IF-0;0PTC COUNT IF-0
P3=(D:2) = FUNCTION CODE SPECIFIED BY DATVER

= 0 => AEAD = 1 => URITE = 2 => CONTROL

=(6:10)= DSTN OF TARGET DATA SEG

### Event -111 (2157)

EVENT HAME: I/O COMPLETION DESCRIPTION: SIO COMPLETION

CALLING MODULE: HARDRES

CALLING PROCEDURE: STODA

PARAMETERS PARAMETER DESCRIPTION

P1=IOQP(DLDEV).LDEVN=LOGICAL DEVICE MUMBER OF DISC INVOLVED IN TRANSFER P2=IODP(DPAR1) (OEFIMED BY DRIVER) P3=IODP(DPAR2) (DFFIMED BY DRIVER)

MMSTAT Event Group 12

Event 120 (2170)

EVENT NAME: SOFT DEATH DESCRIPTION: BUG CRICHER

CALLING MODULE: MARDRES

CALLING PROCECURE: SOFT'DEATH

PARAMETERS PARAMETER DESCRIPTION SOFT'DEATH I.D. HUMBER CALLERS STATUS REGISTER CALLERS DELTA P

Event 125 (2175)

EVENT HAME: IOBUFTAP EVENT DESCAIPTION: IOSYSTEM BUFFER TRAP

CALLING MODULE: HARDRES CALLING PROCEDURE: \$100M

PARAMETER DESCRIPTION

P1 = TOOP P2 = IOOP(QOSTN).DSTN = DST NUMBER OF BUFFER P3 = 0

G.00.00 20- 37

MMSTRT Event Group 14 (CS/3000)

0.00,00 20- 38

MMSTAT Event Group 13

Event 139 (X213)

Event Name: C ABSENT
Description: Either the mapped disc domain or the target
DDST was absent when a cache move was attempted.
Calling Module: CACHESEG
Calling Procedure: PROCESSCOTLDGREQQUEUE

Parameter Description

P1 = D Happed Donain absent P2 = Pin P3,P4 = Segment identifier of Mapped Donain P5,P6 Not used.

P1 = LDR entry index (DST not present)
P2 = Pin
P3,P4 = Segment identifier of DST (P4.(D:1) = 1 stack)
P5,P6 Not weed.

MMSTRTS Events

Event 140 (2214)

EVENT NAME: COPEN DESCRIPTION:

CRULING MODULE: COMSYSZ

CALLING PROCEDURE: COPEN

PARAMETERS PARAMETER DESCRIPTION

P1 (D:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER

P2 PMRP1

P3 PMBP2

### AMSTRIS Events

### Event 142 (%216)

EVENT NAME: CABORTIO DESCRIATION:

CRULING HOOULE: CORSYST

CALLING PROCEDURE: CABORTIO

PARAMETERS PARAMETER DESCRIPTION

P1 LOGICAL DEVICE

A2 IDGINDEX

P3 0

MMSTATS Events

Event 144 (%220)

EVENT NAME: CSIONAIT DESCRIPTION:

CALLING HODBLE: COMSYST

CALLING PROCEDURE: CSICUPIT

PARAMETERS PPRAMETER DESCRIPTION

P1 (0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER

P2 TRANSMISSION LOG

93

Event 146 (X222)

EVENT MAME: COLOSE DESCRIPTION:

CALLING HOODLE: COMSYS3

CALLING PROCEOURE: COLOSE

PARAMETERS PARAMETER DESCRIPTION

P1 (0:\$) = CS ERROR CODE (8:\$) = LOGICAL DEVICE NUMBER

P2 LINE NUMBER

P3 0

G.00.00 20- 41

MMSTATS Events

Event 147 (%223)

EVENT NAME: CREAD DESCRIPTION:

CRELING MODULE: COMSYS4

CALLING PROCEDURE: CREAD

PARAMETERS PARAMETER DESCRIPTION

A1 (0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER

AZ INCOUNT

P3 STATION

Event 149 (XZ25)

EVENT NAME: CHRITE DESCRIPTION:

CALLING MODULE: COMSYS4

CALLING PROCEDURE: CHRITE

PARAMETERS AARRMETER DESCRIPTION

P1 (0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE MUMBER

P2 DUTCOUNT

P3 INCOUNT

MMSTRT Event Group 15 (CS/3000)

9.00.00 20- 42

Event 150 (2226)

MMSTATS Events

EVENT NAME: CSDRIVER DESCRIPTION:

CRILING MODULE: BSCLOM

CALLING PROCEDURE: CSDRIVER

PARAMETERS PARAMETER DESCRIPTION

P1 TIMER LSM

P2 CURRENTSTATE

P3 CURRENTEVENT

WHERE THE DRIVER IS IN THE STATE TERMSITION TABLE (0:8) = CURRENT EVENT (8:8) = LOSICAL DEVICE WHAT CAUSED THE ORIVER TO BECOME ACTIVE

Event 152 (2230)

EVENT NAME: CCONTROL DESCRIPTION

CPLLING MODULE: COMSYSS

CALLING PROCEDURE: CCONTROL

AARAMETERS PARAMETER DESCRIPTION

A1 (0:8) = CS EPPOR CODE (8:8) = LOGICAL DEVICE NUMBER

A2 CONTROL CODE

P3 PARAMETER

### MMSTRTS Events

Event 155 (X233)

EVENT MAME: CPOLLIST DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE: CPOLLIST

PARAMETERS

P1 LOGICAL DEVICE

P2 CS ERROR CODE A3 PMAP

AGRAMETER DESCRIPTION

### Event 153 (2231)

EVENT MAME: COPENTRACEFILE DESCRIPTION:

CRILING MODULE:

CRLLING PROCEDURE: COPENTRRCEFILE

PARAMETERS PARRMETER DESCRIPTION

P1 (0:8) = C5 ERROR CODE (8:8) = LOGICAL DEVICE NUMBER

P2 CTRACEINFO

### Event 154 (2232)

EVENT NAME: CCLOSETRACEFILE DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE: CCLOSETRACEFILE

PARAMETERS

PARAMETER DESCRIPTION

P1 (0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER

P2 0

P3 0

G.00.00 20- 45

### MMSTRTS Events

# MMSTAT Event Group 19

5.00.00 20-46

# Event 191 (2277)

EVENT NAME: DISKINTART DESCRIPTION: A 7905/7920 CONTROLLER IS PROCESSING AN ATTENTION INTERRUPT (ONLINE/OFF.LINE) CRILING MODULE: HARDRES

CALLING PROCEDURE: SIOOM

PRRAMETER DESCRIPTION
(US)--1.e. WHO GOT THE INTERRUPT PARAMETERS Pt= @OITP

P2= BOITP (THEM)--1.e. WHO RRN THE POLL PROGRAM

P3= OITP "OUR" OIT FLAGS WORD

THERE SHOULD BE AT LERST AN X300 AND AN X303 FOR ERCH SIO PROM. A SINGLE ISOLATED (IN TIME) REQUEST WILL DEWERRTE AT LERST A X303, X304, X303. IF THE OUGUE OF IOU'S ON A DIT MEVER EMPTIES, THERE WOULD BE ONE X300 AND ONE X307 PER SIO PROM.

MMSTRTS Events

# MMSTAT Event Group 16

# Event 160 (2240)

EVENT NRME: CREAD DESCRIPTION:

CALLING MODULE: OSMON

CRULING PROCEDURE:

PARAMETERS PARAMETER DESCRIPTION

P1= TIME STAMP

P2= (0:4) NOT USED
(4:1) BLOCK
(5:2) STRET
(7:3) NENT
(10:1) :=0 INITIALIZATION EVENT
(11:5) SUB EVENT NUMBER

P3= OEPENDS ON THE SUB EVENT MUMBER RNO
IF IT IS AN INITIALIZATION OR COMPLETION EVENT.
MSG: (0:4) STRATYPH
(4:5) MSG LS
(10:16) STRATYP

EVENT	INIT	COMP
ME	PRRM	PRRM
ERO	0	LEN
RITE	N MSG	LEN
WAIT	0	LEN
NECK	0	ERRCOO
RTTN	0	0
HC.	N MSG	R MSG
MGEURIT	PRRM	0
INREQ	REQ	Ó
BORT	Ö	T/F
ESET	0	Ó
DATE	R MSG	
PERERO		
	ME ERO IRITE INATIT MECH INT INC INC INC INC INC INC INC INC INC INC	PRRN   PRRN

### MASTRTS Events

### Event 192 (2300)

EVENT NAME: GIPINTERRUPT DESCRIPTION: INTERRUPT JUST PROCESSED

CALLING MODULE: HARDRES

CRLLING PROCEDURE: GIP

PARRMETERS

PARAMETER DESCRIPTION

21 = LDEV

QUEUE ELEMENT WORD ENTRY INDEX P2

РЗ CONTENTS OF DIT HORD O: THE FLAGS HORD

CHANNEL PROGRAM INSTRUCTION POINTER P4

25 CONTROLLER STATUS

P6 = ISU of a Return from TIMER BBSIRIS Events

Event 193 (X301)

EVENT WARE: STARTIO DESCRIPTION: Issuing SIOP machine instruction.

CRELING MODULE: MARGRES

CALLING PROCEDURE: START HPIB, STARTIO

PARAMETERS PARAMETER DESCRIPTION

Absolute address of SIO program to start.

P2 = LOEV number

23 JRI ny⊲ber

# Q'ENTRY'INDEX FROM DITF(DIGGP)

= DIT WORD O: THE DIF FLAGS WORD 25

P6 = LSW of A RETURN FROM A CALL TO TOMER

G.00.00 20-49

MMSTRTS Events

Event 194 (X302)

EVENT NAME: SIDOM-ENTRY DESCRIPTION: Entering SIDOM

CALLING MODULE: HARDRES CRLLING PROCEDURE: SIDOM

PRRRMETERS

PRRRMETER DESCRIPTION

P1 = LDEV

= IDQ OR DRQ table relative index PZ

P3 = DIT WORD O (DIT FLAGS)

= CURRENT STATE OF THE VARIABLE STATE
IN SIDON P4

- UNUSED AT THIS TIME

= LSW RETURNED BY CALL TO TIMER 26

Event 195 (X303)

EVENT NAME: SIDDM-EXIT DESCRIPTION: Leaving SIDDM Haim loop.

CRELING MODULE: HARDRES

CALLING PROCEDURE: SIGON

PRRAMETERS PARAMETER DESCRIPTION

SAME RS EVENT 194 (X302) EXCEPT THAT EVENT IS 195 (X303)

6,00.00 20-50

MMSTRTS Events

MMSIRI Event Graup 20

THESE EVENTS ARE FOR DEVELOPMENT USE ONLY AND ARE NOT NORMALLY ENABLED

Event 200 (X310)

EVENT NAME: DISKBUGGATCHER DESCRIPTION: A HOUNTED VOLUME TABLE CHANGE IS BEING MADE.

CALLING MODULE: PVSYS

CALLING PROCEDURE: MYTABLE

PRRRMETERS PARAMETER DESCRIPTION

P1= FUNCT 0 = DELETE ENTRY 1 = RDD ENTRY 2 = PRESERVE ENTRY

P2= MVTABK (MOUNTED VOLUME TABLE INDEX)

P3= DELTRP (VALUE DF Q-2)

Event 201 (X311)

EVENT MAME: DISKBUGGATCHER DESCRIPTION: A PRIVATE VOLUME USER TABLE CHANGE IS BEING MADE.

CRELING MODULE: PVSYS

CRILLING PROCEDURE: USERTABLE

PARAMETERS PARAMETER DESCRIPTION

P1= FUNCT
0 = CREATE USER ENTRY
1 = REMANE USER ENTRY
2 = RETURN ALL MYTHAK LADITES MOSE SY 4
SPECIFIC POB
3 = RETURN ALL POB POINTERS MOSEMS A SPECIFIC MYTHAK
4 = GET USER ENTRY

P2= MYTREX (MOUNTED VOLUME TRBLE 1906X)

P3= CELTRP (VALUE OF 2-2)

5.00.00 20- 51

0.00.00 20- 52

ANSTRES Events

MMSTRT Event Group 21 Process Creations and

Terminations Logical Process Table

Event -211 (2323)

EVENT NAME: PROCESS COMPLETION DESCRIPTION: PROCESS HAS TERMINATED

CRLLING MODULE: MORGUE

CRLLING PROCEDURE: TERMINATE

PRRAMETERS

PARAMETER DESCRIPTION

P1=0 P2=0 P3=0

MOSTATS Events

MMSIAT Event Group 22

Time Stamp of Event Trace Enable and Disable

Event 221 (₹335)

EVENT NAME: CONFIGURATION INFORMATION DESCRIPTION: EVENT GROUP MASK

CALLING MODULE: CRIG

CRELING PROCEDURE: CONSMON

PARAMETERS PARAMETER GESCRIPTION

P1= MERSMSKG P2= REASTISKT

P3=Reserved

G.00.00 20- 53

MMSTRTS Events

Event 222 (2336)

EVENT NRME: CONFIGURATION INFORMATION DESCRIPTION: MPE VERSION FIX UPDATE

CALLING MODULE: OPCOMMENO

CALLING PROCEDURE: CXMON

PARAMETERS PARAMETER GESCRIPTION

P1= VERSION

P2= FINL

P3= UPDATEL

Event -223 (-2337)

EVENT MANE: CONFIGURATION INFORMATION
OESCRIPTION: SYSTEM TRBLE LICETIONS AND AVAILABLE LIMEO MEMORY
INFORMATION
CALLING MODULE: OPECOMAND

CRLLING PROCEDURE: CXMON

PARAMETERS PRRRMETER GESCRIPTION

P1=F (X1032)=@CST(0)=@OST(0)
=DISPLRCEMENT TO CODE
P2=F(X1033)=@CST(USST)=@OST(0)
=DISPLRCEMENT TO SHRRHBLE
P3=LOGICHL(TOTAL&DLSK(4))=LIMKED MEMORY SIZE

G.30.00 20- 54

MMSTRTS Events

Event -224 -(2340)

EVENT MAME: SYSPINS DESCRIPTION: LOGICAL PROCESS TABLE

CRLLING MODULE: OPCOMMEND

CALLING PROCEGURE: CKNON

PRRRMETERS PARAMETER DESCRIPTION

P1=RBSOLUTE(Z1141)=PROCEN'S PCBENTRY NUMBER
P2=RBSOLUTE(Z1142)=HRN'S PCB ENTRY NUMBER
P3=RBSOLUTE(Z1143)=UCOP'S PCB ENTRY NUMBER

Event -225 (-2341)

EVENT NRME: SYSPINS(CNTD.)
DESCRIPTION: LOGICAL PROCESS TRBLE

CALLING MODULE: OPCOMMEND

CRLLING PROCEDURE: CXHON

PRRRNETERS PPRAMETER DESCRIPTION

P1=RBSOLUTE(X1144)=PFRIL'S PCB ENTRY NUMBER
P2=RBSOLUTE(X1145)=OEVREC'S PCB ENTRY N
P3=RBSOLUTE(X1146)=PRMSG'S PCB ENTRY N

Event -226 (-2342)

EVENT NRME: SYSPINS(CNTD.)
DESCRIPTION: LOGICRE PROCESS TABLE

CRLLING MODULE: GPCOMMAND

CALLING PROCEGURE: CHOON

PARAMETERS PARAMETER GESCRIPTION

P1=ABSOLUTE(X1147)=STMSG'S PCB ENTRY # P2=ABSOLUTE(X1150)=LOG'S PCB ENTRY # P3=ABSOLUTE(X1151)=LOAD'S PCB ENTRY #

### MMSTATS Events

Event -227 (-2343)

EVENT NAME: SYSPINS(CNTO.)
OESCRIPTION: LOGICAL PROCESS TABLE

CRLLING MODULE: OPCOMMANO

CALLING PROCEDURE: CXMON

PARAMETERS PARAMETER DESCRIPTION

P1=ABSOLUTE(X1152)=IONESSPROC'S PCB ENTRY #
P2=ABSOLUTE(X1153)=SYSIOPROC'S PCB ENTRY #
P3=RBSOLUTE(X1154)=HENLOGP'S PCB ENTRY #

Event -228 (%344)

EVENT NAME: TIMESTAMP DESCRIPTION: TIMESTAMP

CRLLING MODULE: DPCDMMRND

CRLLING PROCEOURE: CXMON

PARAMETERS PARAMETER DESCRIPTION

P1=CRLENORR (0:7)=YEAR OF CENTURY (7:9)=DRY DY YEAR P2=CLOCK(WORD1). (0:7)=HOUR OF ORY (8:8)=HINUTE OF HOUR P3=CLOCK(WORD2). (0:7)=SECDNOS INTO HIMITE (8:8)=TENTHS OF SECONOS

Event -229 (-X345)

EVENT NAME: MONOFF OESCRIPTION: END EVENT TRACING

CALLING MODULE: DPCOMMRNO

CRLLING PROCEOURE: CXMON

PRRRMETERS PRRRMETER DESCRIPTION

P1=0 P2=0 P3=0

G.00.00 20-57

MMSTRTS Events

MMSTRT Event Group 23 (Terminal I/O)

Event 230 (%346)

EVENT NRME: TERMREAD
DESCRIPTION: TERMINAL RERO COMPLETION

CRLLING MODULE: HARDRES CRLLING PROCEDURE: TIP

PARAMETERS

PARRMETER DESCRIPTION

P1 = LDEV P2 = RERO OURRITION P3 = SYTES RERD

Event 231 (%347)

EVENT NRME: OCIDC2RCK
OESCRIPTION: OCI/OC2 HAS BEEN SRTISFIED

CALLING MODULE: HARDRES CALLING PROCEDURE: TIP

PARAMETERS

PARAMETER DESCRIPTION

P1 = LOEV P2 = DURATION (BETHEEN START AND DC2) P3 = BYTES READ (EXCLUDING DC2)

G.00.00 20- 58

PARAMETER DESCRIPTION

PARAMETER DESCRIPTION

MMSTATS Events

Event 232 (%350)

EVENT NAME: TERMURITE
DESCRIPTION: WRITE COMPLETION

CALLING MODULE: IOTERMO CALLING PROCEDURE: TERMION

PRRRHETER DESCRIPTION

P1 = LDEV

P2 = 0 P3 = BYTE COUNT OF TRRNSFER

Event 233 (%351)

EVERT NAME: BINRERO DESCRIPTION: BINRRY READ COMPLETED

CALLING MODULE: HARDRES CALLING PROCEDURE: TIP

PARAMETERS

PARAMETER DESCRIPTION

P1 = LDEV P2 = OURATION P3 = BYTES READ

MMSTATS Events

Event 234 (X352)

EVENT NRME: TERMLOGON
DESCRIPTION: TERMINRL JUST LOGGING ON

CALLING MODULE: INTERMO CALLING PROCEDURE: TERMION

PARAMETERS P1 = LDEV P2 = 0 P3 = 0

Event 235 (%353)

EVENT NAME: TERMLDGOFF
DESCRIPTION: TERMINAL JUST LOGGED OFF

CRLLING MODULE: IOTERMO CALLING PROCEDURE: TERMIOM

PARAMETERS

G.00.00

Event 240 (2360)

### Event 236 (2354)

EVENT WAME: SPECCHAR
DESCRIPTION: PROCESSED SPECIAL CHARACTER

CALLING MODULE: HARORES CALLING PROCEDURE: TIP

PARAMETERS

PARAMETER DESCRIPTION

P1 = LDEV
P2 = SPECIAL CHAARCTEA PROCESSEO
P3 = 0

### Event 237 (2355)

EVENT NAME: BREAK DESCRIPTION: PADCESSED BREAK

PARAMETER DESCRIPTION

PARAMETERS P1 = LDEV P2 = DSTATE P3 = D

### Event 238 (2356)

EVENT NAME: SPECAERO DESCRIPTION: SPECIAL READ TERMINATION CHARACTER DETECTED

CALLING HODULE: HAROAES CALLING PROCEDURE: TIP

PARAMETERS

PARAMETEA DESCRIPTION

P1 = LDEV P2 = DURATION P3 = BCNT

## MMSTAT Event Group 24 (Power Fall)

Event Name: PFAIL
Description: Power fail detected.
Calling floodule: ININ, PFAIL
Calling Procedures: Powerup (ININ), Powerup (PFAIL)

# Parameter Description

P1 = 0 Called from Pouerup in ININ 1 Called from entry in Pouerup in PFRIL 2 Called from end of Pouerup in PFRIL

P2 = For P1=0 this is 0 For P1=1,2: TAUE = Multiple powerfail FALSE= First powerfail

P3 = PF
D = No powerfall or PFRIL processing complete
1 = Set by the power down trap in ININ
2 = Set by the power up trap in ININ
3 = Set when awake the PFRIL process
4 = Set by PFRIL after message appears on console

P4 = SYSUP 0 = System not back up after powerfall 1 = System back up after powerfail

P5,P6 not used.

G.00.00 20- 61

6.00.00 20- 62

### CHAPTER 21 AGOTFILE LAYDUT

### General Rootfile Layout

LABEL O | ROOTFILE INFORMATION PASSUOAD TABLE PASSWOAD TABLE (CONT.) 3 ITEM R/W TABLE

SET A/N TABLE

RECORD O OATABASE GLOBAL INFO 128 uds ITEM TABLE (variable size) (variable size) DATA SET CONTAOL BLOCKS (DSCB) (variable size)

The data base RDDT FILE is an MPE file with filecode equal to -400. The record size is 128 words, fixed, binary format with a blocking factor of 1. The size of the file depends on the number of data items and data sets defined in the data base.

G.00.00 21- 1

### Root File Label O

		Z
MORD O	AL'CONDITION (rootfile condition)	0
1	RL'DATE(creation_date)	1
2	RL'IINE (creation time)	1 2 3 4 5 6 7
3 [		3
4	RL'EVEROPEN	4
5 I	_RL'COLDLOADID(cold_load_id)	5
6	_RL'USERCOUNT	6
7 [	_RL'DBCBDSTNUN(DST_number_of_DBCB)	7
В	RL'LDGID (log id for	10
. 1	transaction logging)	
. 1	·	
11 [		13
12 [	AL'LOGPASS (log id password)	14
. 1		
		• -
15 (		17
	AL'FLAGS (database_flags)	20
	RL'STORDATE (DBSTORE_date)	21
18	AL'STOATIME (DBSTORE time)	22
19 (		23
20	AL'BUFSPECCOUNT(buffer_spec_count)	24
21	RL'ILRCAEATEOATE (date ILR log_created)	25
22	RL'ILRCREATETIME (time ILR log created)	26 27
23   24	RL'ILRURSTDATE (last log access date)	30
25	RL'ILRLASTTIME (last log access time)	31 32
26   27	RESERVED	33
21	FOR .	23
	FUTURE	٠
63	USE I	77
64	RL'MAINTHOAD (database maintenance	100
04	Hord)	100
•	, , , , , , , , , , , , , , , , , , ,	
67		103
68	RL'BUFFERSPECS (buffer specifications)	104
to		
	i	
127	i	177

AL'CONDITION (IN ASCII):

JB - Virgin. The database has not been created yet.
FU - OK. The database is DK.
All - Modified deferred. The database is being modified.
NC - Maintenance create. The database is being created.
NE - Maintenance erase. The database is being created.
IL - ILR recovery in progress.

G.00.00 21- 2

## General Rootfile Lavout

# Agot File Label 0 (cont.)

RL'DATE: Acct file creation date\*. Its format is:

\_0:\_1:\_2:\_3:\_4:\_5:\_6:\_7:\_B:\_9:10:11:12:13:14:15 |year\_\_\_\_\_|day\_of\_year\_\_\_\_

AL'TIME: Root file creation time\*. Its format is:

0: 1: 2: 3: 4: 5: 6: 7: 8: 9:10:11:12:13:14:15
|hour | hinutes | tenth\_of\_seconds |

AL'EVEROPEN: This field is no longer used under IMAGE B

AL'FLAGS:

AL'STDRDATE: Same format as RL'DATE\*.

AL'STORTIME: Same format as RL'TIME\*.

AL'BUFSPECCOUNT: Maximum number of buffer specifications allowed.

RL'ILREREATEDATE: Same format as AL'DATE\*.

RL'ILRCREATETIME: Same format as AL'TIME\*.

AL'ILRLASTDATE: Same format as RL'DATE\*.

RL'ILRLASTTIME: Same format as RL'TIME\*.

RL'MAINTHORD: For data bases with no maintenance word this field has 2 semicolons (';;') and trailing blanks.

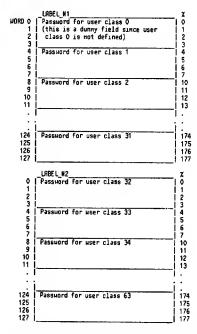
### General Rootfile Lavout

# RL'BUFFSPECS:

BIT/	0: 1: 2: 3: 4: 5: 6: 7: 8: 9:10:11:12:13:14:15	X
ND 68	buffers for 1 user   buffers for 2 users	104
69	buffers for 3 users   buffers for 4 users	105
	etc	
	ll	
127	buffers_for_119 users_ buffers_for_120 users_	177

\* The DATE and TIME fields can be formatted (for display purposes) individually by calling the FNTCHERDAR and FNTCLOCK Intrinsics respectively. Dr both fields can be formatted at once with FNTDATE Intrinsic.

### Root File Labels 1 & 2

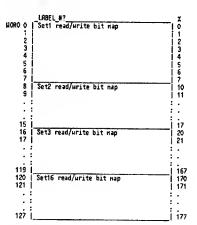


The PRSSMORD TABLE occupies user labels number 1 and 2. There are four words (8 characters) reserved for each password. The relative position of a password corresponds to the user class number defined in the schena. For user class numbers not defined in the schena. For user class numbers not defined in the SCHERR, the four word field is filled with blanks.

G.00.00 21- 5

General Rootfile Lavout

## Root File- Next Label



The SET RERD/URITE TRBLE starts on a user label boundary after the ITEM RERD/URITE TRBLE.

There are eight words for each SET RERD/URITE bit map.

There are eight words for each SET RERD/URITE bit map.

There are eight words for each SET RERD/URITE bit map.

The databases with more than 16 data sets, the read/urite table continues in the next user labels.

The specific format of this table is shown in the next page.

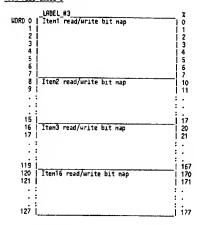
The number of user labels occupied by the SET RERD/WRITE TRBLE depends in the number of data sets defined in the schema, and is obtained by rounding upwards (ceiling) the result of:

Nun-of-labels = [(Nun-of-sets)\*8]/128

 $\text{Max-size} = ((99)*8)/128 = 6.18 \Rightarrow 7 \text{ labels}$ 

General Rootfile Layout

Root File Label 3



The IIEM RERD/WRITE TABLE starts in user label #3
There are eight words for each ITEM RERO/WRITE bit nap.
For databases with more than 16 items, the read/write table continues
in the next user labels. The specific format of this table is explained
after the SET RERD/WRITE TABLE since it is defined the same way.
The number of user labels occupied by the ITEM RERO/WRITE TABLE depends
on the number of data items defined in the schema and can be obtained
by rounding upwards (ceiling)the result of:

Num-of-labels = [(Num-of-itens)\*8]/128

Since there can only be a naximum of 255 data items in the schema, the maximum size for this table in user labels would be:

Max-size = [(255)\*8]/128 = 15.93 => 16 labels.

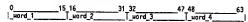
G.00.00 21- 6

General Rootfile Layout

### Iten/Set Read/Write Table Format

There are eight words per item/set read/write table definition and up to 16 items/sets per record (user label). Within each 8 words, the first 4 words are the flags for the user classes which have read access to the item/set. The second 4 words are the flags for the user classes which have write access to the item/set. The detail format for an eight word field is shown below.

A. Four words for read access:

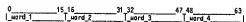


4 words represent 64 bits. Bit n represents read access for user class n to the iten/set. If bit n is set to 1 then user class n has read access to the iten/set. For example, if the word settings are:

могd 1 могd 2 могd 3 могd 4 2000016 2020000 2000410 2001300

This means that user classes 12, 13, 14, 18, 39, 44, 54, 56 and 57 have read access to the item/set. If no read/urite security is defined at all for the item/set, then all of the read security bits are set to 1.

B. Four words for write access:



Write access flags have the same format as the read access flags. Bit n represents write access for user class n to the iten/set. If bit n is set to 1, then user class n has write access to the iten/set For example, if the word settings are:

могd 1 ногd 2 могd 3 ногd 4 2000010 2020000 2000000 2001100

This means that the user classes 12, 18, 54 and 57 have write access to the item/set. If no read/write security is defined at all for the item/set, then all of the write security bits are set to 0.

### Root File Record 0

RECDRD #0
HORD | \_RODT'DBSTATUS
1 | RDDT'DBNAME 4 5 6 7 10 11 12 13 14 15 17 NDUDPEN 20 21 22 MAXDPEN RESERVED (for future use) 18 (set to binary Os) 127 177

RODITORSTATUS

(0:8) - IMAGE version ('B' in ASCII) (8:8) - Binary 1 (filler)

RODT'OBNAME - DATABASE name left justified (last 2 chars are blank).

NOWDPEN - Number of data sets opened. This field is not used in IMAGE B

MAKOPEN - Maximum number of data sets that can be opened. This field is not used in IMAGE 8.

NDTE:

RDOT'ITEMPTR and ROOT'DSETPTR is a word offset from record 0 (be-ginning of the file, not including the space taken by the user la-bels) and can span several records.
These pointers point to the Oth entry of the table and since the Oth entry in the item table or the set table does not really exist, they actually point to 11 words before the beginning of the table. To get to the first entry in the table, this pointer should be incremented by the length of the entry (which is currently 11 words).

G.00.00 21- 9

General Rootfile Layout

Root File- Next Record

\_0:\_1:\_2:\_3:\_4:\_5:\_6:\_7:\_8:\_9:10:11:12:13:14:15 | set-name-1 reserved-1 |\_data-set-type set-no-of-synonyH\_ \_reserved-2\_ DSCB-pointer 10 12 13 11 12 13 14 15 16 17 18 19 20 21 22 14 15 16 17 20 21 22 \_\_\_\_reserved-1\_\_\_ \_\_data-set-type 23 24 25 26 set-no-of-synonyn\_ \_reserved-2\_ \_\_DSCB-pointer

Set table follows the Item table.

Each entry is 11 words long. The length of the table depends on the number of data sets defined in the schema. The relative position of a set definition depends on its relative position in the schema.

Set-mane: is a data set name, left-justified and with trailing blanks.

Set-number-of-synonym: is the number of a data set whose name has the same hashed result as this one (this is utilized for quick set name searches).

Oata-set-type is one of the following: A, M or D.

DSC8-pointer: is a pointer to the Data Set Control Block. This pointer is word offset from record #0. The DSC8 is described ahead.

The maximum size for this table is 11\*99 = 1089uds.

NOTES: The reserved-1 and reserved-2 fields are the 'old' level numbers for the read and write access respectively. Since this concept no longer applies, the values are set to zero.

General Rootfile Lavout

Root File Record 1

```
10
11
12
13
14
15
16
17
20
21
           reserved-2
                                       |_1ten-type_
|_sub1ten-length
     10
         |_subiten-count
     11 | iten-name-2
     12
13
14
15
16
17
     18
19
20
21
                                                                      22
23
24
25
26
         _item-no-of-synonym_
_reserved-2
                                       |_1ten-type_
|_subiten-length
          _subitem-count
     22
```

The ITEM TRBLE starts in record #1. Each entry is 11 words long and the length of the table depends on the number of data items defined in the schema. The relative position of an item definition depends on its relative position in the schema.

Item-name: is a data item name, left-justified and with trailing blanks

Item-number-of-synonym: is the number of the item whose name has the same hashed result as this one (this is utilized for quick item name searches).

Item-type: is one of the following: I, J, K, R, X, U, Z, or P

iten-type VALUES, 20J2: | |subitem-length |subitem-count

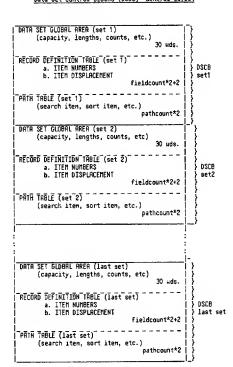
The maximum size for this table is 11\*255 = 2805wds

The reserved-1 and reserved-2 fields are the 'old' level numbers for read and write security. Now, the values are always zero.

G.00.00 21- 10

General Rootfile Layout

Data Set Control Blocks (DSCB)- General Layout



The DSCBs follow the SET TRBLE in the Root file. There is one DSCB for each data set defined. The function of the DSCB is to define each data set within the data base.

G.00.00 21- 12

### Data Set Control Block (Global Rrea)

DSBLDCKLGTH (block_length) DSBLDCKLGTH (redia_record_length) DSSRLDCKFRC DSFIELDCT
3   DSMEDIALOTH
DSENTRYLOTH (entry length)
DSFIELDCT DSFIELDCT
DSPRIMEY X DSPRIMEY
DSPATHPIR(offset to path table)
logical end of file
1   *
Nax num of records in set
18 words of binary zeros
,

DSCRP - data set capacity as reported by the SCHEMR processor.

DSBLOCKLGTH - data set block length including the bit map overhead.

DSMEDIRLGTH - data set media record length (remember that this length includes the pointer overhead)

DSERTRYLGTH - data set entry length.

DSBLDCKFAC - data set blocking factor.

 $\begin{tabular}{lll} {\tt DSF1ELDCT} & - \mbox{ data set field count. This is the number of fields} \\ & \mbox{ specified for the data set.} \end{tabular}$ 

DSPATHCT data set path count. This is the number of paths that are specified for the data set.

X-DSKEYTYPE - data set key type. If DSKEYTYPE = TRUE then the key is hashed.

DSPRIMKEY - data set primary path or key.
For master data sets, this is the field number of the

search item.

For detail data sets, this is the field number of the

DSPRIMPIR - data set path table pointer. Word offset to the data set path table which contains an entry for each path defined. It points to path Oth entry in the table, so to get to the first entry the pointer should be incremented by the length of the entry (which is currently 2 words).

G.00.00 21-13

General Rootfile Layout

### Data Set Control Block (Item Numbers)

^	D:_1:_2:_3:_4:_5:_6:_7: item_num_of_1st_field_ item_num_of_3rd_field_	_8:_9:10:11:12:13:14:15
MOLE	_1TeM_nuH_0f_18T_f181d	1 Item num of 2nd field
1	_item_num_of_3rd_field_	etc.
	etc.	binary D
	_binary_0	_binary_D

The Item Numbers Table follows the Global Rrea of the OSCB. The size of this table (in words) is equal to the number of items in the given data set plus 1. The first n bytes are used to carry the item numbers of the fields within the data set. The remaining n+2 bytes are set to binary zeros.

# Data Set Control Block (Record Definition Item Displacement)

Hord D	_D:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15  _word_offset_to_first_field
1	_uord_offset_ta_second_field
2	_uord_offset_to_third_field
	:
	:
•	_word_offset_to_last_field  _length_of_entry

This table immediately follows the Item Munbers Table.

The word offset points to the starting location of the field within the media record. Remember that the media record includes the pointer overhead so this offset varies for master and detail data sets: if a master data set has only one path, the word offset for the first field is 10, since there are 10 words of overhead-5 words for the synonyn chain pointers and 5 words for the data set chain head that it would be pointing to. On a detail data set with one path, the overhead is only 4 words.

The 'length-of-entry' field is the same as the media record length.

G.00.00 21- 14

### General Rootfile Layout

### Data Set Control Block (Path Table)

	_D:_1:_2:_3:_4:_5:_6:_7:_B:_9:10:11:12:13:14:15   1st path definition
Hord C	1st path definition
2	2nd path definition
4	
	last path definition

There are 2 words (4 bytes) for each path definition.
The PATH TRBLE for master data sets has a different layout from the
PATH TRBLE for detail data sets.
Baster sets:
Byte Description
1 - iten number of the search iten in the related
detail set.

- 1 iten number or the search atom and detail set.
  2 iten number of the sort iten in the related detail set.
  3 set number of the related detail data set
  4 path number of the corresponding path in the related detail data set.

Detail sets:

- Detail sets:

  Byte Description

  1 field number of the search item.

  2 field number of the sort item.

  3 set number of the related naster data set

  4 path number of the corresponding path in
  the related master data set.

### General Data Set Layout

Word	D-1	USER_LABEL_O nasters=capacity details=highwater mark
Hord	2-3	nunber of unused records
Word	4-5	masters= mot used details= delete chain head

# General Rootfile Layout

Record	0	RECORD D through n data records	
Record	n		!

# Data Set User Label O

Word D-1: Record name of the highest readable record. for Masters, this is the highest record in the set (i.e. Capacity). For Details, this is the greatest number of records that have been written to the set thus far. For example, if there is roon in the Detail data set for 100 records and 75 were written last week when the data set was loaded with DBLDMD, and yesterday 15 records were deleted from the data set, the "High Water Mark" is equal to a value of '75'.

Word 2-3: Number of unused records in the data set. This field is incremented when a record is deleted and decremented when a record is added. To determine the current number of entries used in the set subtract Word 1-2 (unused court) from United 1-1 (capacity) Word D-1 (capacity).

Word 4-5: The delete chain head for Details. This points to the record most recently deleted or contains a value of zero if no records have been deleted. This field is not used in Master data sets.

# Data Set Records

The data in the data set records is arranged according to the Media records. These are formatted by the Schema Processor (DBSCHEMR).

### CHAPTER 22 DISC FREE SPRCE MAP

### Disc Resident Data Structures

There are two disc resident free space data structures, the bit map and the descriptor table, for each disc volume that has a free space map, i.e. system discs and private volumes. The addresses of these data structures are kept in the disc label. The symbols that define the descriptor table and bit map are in the include file INCLDFS2.

### Bit Map

The bit map is divided up into pages, which is the physical block of the map that is read or written. At the moment, a page is defined to be one sector (128 words) long, this may be changed by changing a compile time constant. The last word of the page is a checksum for that page, all other words are data. There is a one to one correspondence between bits in the map and sectors of the disc. A one bit represents a free sector and a zero bit represents an allocated sector. The bit map is a contiguous set of pages, enough to represent the entire disc, excluding spare tracks and spare sectors.

### Descriptor Table (DT)

The descriptor table is an array of three word entries, one entry for each page of the bit nap. Each entry looks like this:

= largest space = unnd 0 -----= starting space = uord 1 word 2 = ending space = -----

Thus the descriptor table looks like this.

= entry for page 0

G.00.00 22-1

Disc Free Space

= entry for page 1 = entry for page 2 = entry for page 3 = entry for last page

Each entry describes the free space on the corresponding page of the bit map. The largest space und is the size of the largest contiguous block of free space on the page, which is not at the very beganning or very end of the page. That is, the first bit physically representing the space is not the first bit of data on the page or the last bit representing the space is not the last bit of data on the page. Starting space is the number sectors of contiguous space represented by the set of bits whose first bit is the first bit of data on the page. Ending space is the number of sectors of contiguous space represented by the set of bits whose last bit is the last bit of data on the page. The starting space and ending space fields allow looking across page boundaries, thus preventing fragmentation on page boundaries. Thus, if all sectors represented on a page are free, then starting and ending space uill be the same and have the total number of free sectors represented on the page. Largest space will be zero, as there is no block of space that is not at the beginning or end of the page. R value of -1 for all the fields in an entry indicates the corresponding page is bad, either from a checksum or I/D error.

### Virtual Memory Resident Data Structures

For each system disc or physically nounted private volume there is a data segment which has information about the disc free space map, the current copy of the descriptor table, some work space for the procedures while in split stack mode and buffers for pages of the bitmap. The DSI number of the data segment for a given disc is found in the LDTX entry for that disc.

### Disc Free Space Data Segment

For each system disc or physically mounted private volume in the up and running system there is a DST which contains information about the disc free space map for that disc, some work area, a copy of the descriptor table ambuffers for the pages of the bit map. All symbols that define these data segments are in the include file INCLDFS1, and they are prefixed with "ds'". The structure of the data segment is as follows:

0 (X0) = de<sup>1</sup>1deo

G.00.00 22- 2

Disc Free Space

	=
1 (%1)	= ds'dst =
2 (%2)	= = = = = = = = = = = = = = = = = = =
3 (%3)	= = =
4 (%4)	= ds'last'page'of'nap =
5 (%5)	= ds'last'buffer'index =
6 (%6)	=================================
7 (27)	= ds'nap'address= = =
8 (210)	= ds'lock =
9 (211)	= ds'lock'count =
10 (212)	= ds'queue'head =
11 (%13)	= ds'queue'tail =
12 (%14)	= ds'descriptor'table =
13 (215)	= ds'buffer'page'number =
14 (216)	= ds'buffer'dirty =
15 (217)	= ds'buffer'area =
16 (218)	= ds'first'threshold'page =
17 (721)	===================================
	= ds'size'of'last'allocation=
18 (%22)	: :

Disc Free Space

19 (%23)	= ds'last'page'allocated'fron =
20 (%24)	= ds'next'buffer'index =
21 (%25)	= ds'page'nunber =
22 (%26)	= ds'word'number =
23 (%27)	= ds'bit'nunber =
24 (230)	= ds'page'pointer =
25 (%31)	= ds'starting'word'number =
26 (%32)	= ds'starting'bit'number =
27 (%33)	= = = = = = = = = = = = = = = = = = =
28 (%34)	= = = =
29 (%35)	= ds'b1t'count =
30 (236)	= ds'entry'type =
31 (%37)	= ds'buffer'index =
32 (240)	± = 4.1322.334
33 (X41)	= ds'disc'address
34 (142)	= ds'error'status =

The rest of the data segment contains tables whose size and location is dependent on the size of the disc and or the number of buffers in the data segment. They are shown below just to demonstrate there relation to one another, for there actual location, the pointers should be examined. The symbol "ds'array" area" defines the start of the area. The first table is the descriptor table, it is in the same format as the disc copy, but a dumny entry of all zeros is added before and after the table, these are needed by procedures "Find'Page" and "Bulld'Descriptor"Entry". The pointer to this table is "ds'descriptor'table", it points to the entry for page zero, not the discrepance of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the size of the s dunny entry.

= 0 =	
* 0 *	dunny
= 0 =	entry
***************************************	
= largest space =	
starting space =	entry for
* ending space *	page 0
<pre>= largest space =</pre>	
= starting space =	entry for
ending space =	page 1
<pre>ending space =</pre>	-
<u>:</u>	
:	
= largest space =	
= largest space =	entry for
= starting space =	•
	last page
ending space	
= 0 =	
	dunny
= 0 =	_
= 0 =	entry
***************************************	

The next table is ds'buffer'page'number table, it has a one word entry for each buffer in the data segment. Each entry contains the page number of the page currently in the corresponding buffer or -1 if the buffer is empty. This is pointed to by "ds'buffer'page'number".

======		==		========
=	buffer			=
=====		==	22222	
=	buffer	1	entry	-
******	********	==		
		:		

G.00.00 22- 5

a last buffer entry a

The next table is the de'buffer'dirty table, which has a one word entry for each buffer. A TRUE indicates the page in the corresponding buffer is dirty, i.e. the disc copy is not up-to-date. A FRISE indicates that the buffer is clean. If DFS was compiled with dirty buffer namagement turned off, this table is not present and the ds'buffer'dirty pointer is zero.

20000		
=	buffer 0 entry	,
=====		
=	buffer 1 entry	
======		====:
	:	
	:	
	:	
=====	=======================================	====:
=	last buffer entry	

The remainder of the data segment contains the buffers, each buffer is the size of one page of the bit map, which is currently one sector(128 words). The beginning of the buffer area is pointed to by "ds'buffer'area" and the number of buffers is the value in "ds'last'buffer'index" plus one.

	=======================================
=	
=	
=	
=	buffer 0
=	
=	
=	
=	
=	
=	
=	buffer 1
=	
=	
=	
**********	

G.00.00 22- 6

Disc Free Space

	:
	:
=======================================	
=	2
•	=
<b>T</b>	=
= last	buffer =
=	
=	=
=	

Each of the fields of the data segment is described in the include file INCLOFS1, where they are defined. It should be noted that the following fields are just workspace, used to pass information between procedures while in split stack node and have no meaning between calls to the disc free space management subsystem:

ds'page'number ds'bit'number ds'starting'uord'number ds'number'of'sectors ds'bit'count ds'disc'address ds'uord'number ds'page'ptr ds'starting'bit'number ds'entry'type ds'buffer'index

The field ds'error'status normally has no meaning between calls unless the error'type field has a value greater than "fatal'dfs'error", in which case it means that disc space may no longer be allocated on this disc.

### CHAPTER 23 MPE DISC CACHING

### Diec Caching Overview

Disc Caching is an optional feature of MPE that utilizee excess main memory and excees CPU horsepower to keep portions of frequently referenced disc "donains" in memory. (A disc "donain" is a copy of a portion of disc residing in main memory. These disc donaine are considered "cached" when they are in memory and are considered "happed" when there is I/D pending against them.) Disc Caching manages the bi-directional transfer of these disc donains between main memory and disc storage. No main memory is permanently dedicated to cached disc donains. Cached disc donains share main memory with all other types of MPE segments and are not treated differently by the memory manager. By keeping cached disc donains in memory, a significant portion of the references to disc etorage can be resolved without actually having hysically access the disc. Disc Caching policies are integrated into the MPE Kermel, File System, and I/D System which allows the system performance to be tuned based on the current workload and resource availability.

Disc Caching uses the MPE kernel resource management mechanisms and strategies. These mechanisms are extended to handle cached disc domains in the eame manner as segments. Thus, cached disc domains can be of variable size, fetched in parallel with other segments or cached domains, garbage collected, and replaced in the same manner as stacks, data and code segments. The relative use of main memory between stacks, data and code segments, and cached disc domains is dymanic. This partitioning is based on the workload's current requirements and current memory availability.

Disc Caching can be enabled/disabled on a disc by disc basis. When caching is enabled for the first disc, the code segment containing the Disc Caching code will be locked into menory. Also at this time the Cache Directory Table (CDT) will be built and locked into menory. When caching is disabled for the last disc, the code segment uill be unlocked from menory and the CDT will be released. Thus if caching is not enabled no memory will be wasted.

The CDT is used to keep track of the following information:

- The disc Idevs currently enabled for caching. There will be a Device Entry in the table for each cached disc.
- A linked list of cached domains for each disc with caching en-abled. The head and tail of this linked list will be contained in the Device Entry. (I.e. there is a separate linked list of cached domains for each cached disc ldev.)
- 3) The cached domains that currently have user I/O pending (i.e. FRERDS/FURITES) or have memory management I/O pending (i.e. fetching the disc domain into memory, or posting the disc domain back out to disc). There Hill be a Mapped Domain Entry in the table for each disc domain has that I/O pending and is thus "mapped".

Disc Caching

ordered writes include things like updating disc free space maps for a new file extent before updating the file extent map in the file label.

There are two disc request entries used for disc caching requests. The first entry is a Logical Disc Request (LDR) entry and is used to nanage the data moves to/fron the user's data area and the disc domain (i.e. the logical I/D). The second entry is a regular Disc Request (DRQ) entry and is used to perforn the physical J/D necessary to map a disc domain (for a read "miss") or to perforn the physical post (on write requests). The disc domain "lerean mapped until both the logical and physical I/D completes. If a request is not completely described by one disc domain already in memory or a Mapped Domain CDT entry (i.e. the requested disc area Falls into nore than one disc domain, then the overlapping disc domain(s) will be flushed to disc and the neu complete disc domain will be fetched (if read) and napped - no partial mappings are allowed. nappings are allowed.

The DST number of the Cache Directory Table (CDT) is at %1273 and the bank and offset are kept in %1274-%1275. The Caching Sir (2) is used when starting and stopping caching (via :STRATCREH:/STOPCREH) and by the LDRDER when loading a program file (this air is only used when updating the STT at load

When caching is enabled for a disc, a bit in the flags word of the DII is set. Also, the Global Serial Write queue can be found by examining the header entry of the Disc Request Table. See Chapter 13 for a more detailed explanation of both the DII and the Disc Request Table header. See Chapter 2 for a description of the Memory Region Meader for a disc domain (cached region).

Diec Caching

4) A linked list of all user I/O pending against the mapped diec domains. There will be a logical Disc Request (LDR) queued to the Mapped Domain entries that will describe the user I/O to take place. This is analogue to a Disc Request queued to a specific OIT waiting for service.

When a request is made to access disc information, Disc Caching must first determine if the requested disc donain ie present in memory. Disc Caching will first determine if the requested area of disc is already mapped into memory by scanning through the Napped Donain entries of the CDT. If the requested transfer can be eatisfied with a currently mapped diec donain, then the I/D request will be queued (FIFD) behind the other I/Ds pending against that napped donain. If the requested area is not already mapped, then a search ie made through the linked list of cached diec donains for the specified disc ldev. (The region header containe the diec addrece and eize that a diec donain represents.) If the requested domain ie found in the list (i.e. present in memory), then this region will be mapped. R donain is then considered mapped when there is an entry for it in the Napped Donain portion of the CDT. Napping the domain allows Diec Caching to manage the I/D pending and/or currently active for a particular disc donain. Dnce the disc donain is mapped and present, the data can be noved between the process' data area and the mapped disc donain. The process can then continue executing without interruption or a process switch. The user/subsystem process for which the move is done will be charged with the CPU overhead.

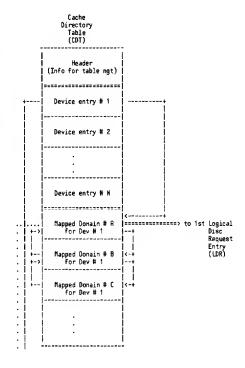
When a request is made to read data that is not currently cached in menory (i.e. a read "miss"), the fetch strategy uses the file System's knowledge of the type of access (sequential or randon), the extent size of the file, along with the current menory load to select the optinal size of the disc domain to be fetched and napped into menory. The fetch of the disc domain is then initiated on the user's stack without a process switch. After the fetch ie initiated, it completes in an umblocked manner so that this process (if no-mait I/O) or another process can proceed in parallel with the cache fetch.

In general, when writing, a process will not wait for completion of the physical I/D. Instead, the process will be awakened as soon as the transfer has completed between the process's data area and the napped disc domain (i.e. no-wait-for-post). The physical I/D will then be posted at background priority while the process continue. (Users can specify wait-for-post on a file by file basis in place of the default no-wait-for-post with the FSETMODE intrinsic. This can be done on a global basis via :CRCHECONTROL.) If the access request is a write and there is a current write pending against the specified napped disc domain, the process request is queued until the pending write is posted to disc. If the disc domain to be written is not currently cached in menory, a free piece of menory will be obtained to nap the corresponding disc inage and then the "write" takes place from the process' data area to the napped disc domain. This prevents data from having to be read before being written. After that, a post to disc is initiated (on any write only the portion of a napped disc domain is complete and the post disc is unitiated, the process performing the "write" is allowed to continue to run without having to wait for the post to complete. Writes that must be posted to disc in a certain order use the Global Serial Write Queue. These

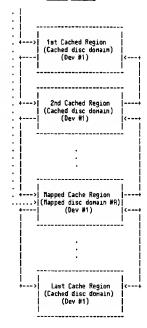
G.00.00 23- 2

Disc Caching

Disc Caching Tables Dverview



### Menory Regions



Cache Directory Table

<u>CDT Neader Entry</u>
This entry contains all information necessary to manage the entire table and also contains global caching related information.

<u>CDT Device Entry</u>
There will be one of these entries for every disc idev that currently has caching enabled. These entries keep track of all cached disc domains in memory for this device. In addition, these entries contain statistics regarding the number of I/Ds performed to the idev.

<u>CDI Mapped Donain Entry</u>
These entries describe disc domains that are currently "mapped" into memory. This means that there is logical I/D (cache nove) and/or physical I/D (fetch or post) pending. These entries keep track of the state of the cached disc domain (IMI, RDC, etc.) just as the DSI Table keeps track of data segments.

The following low core cells contain the address of the CDT:

X1273 contains the DST Number of the CDT
X1274 contains the Bank Number of the CDT
X1275 contains the Dffset within the bank of the CDT

6.00.00 23- 5

## Disc Caching

# Header Entry

0	# Entries	CDT'ENTRIES
1	Entry Size (X30)	CDT'SIZE
5	# Free Entries	CDT'FREE'CDU#T
3	1st Free Entry (table offset)	CDT'FREE'NERD
4	Last Free Entry (table offset)	CDT'FREE'TRIL
5	Max # Entries Used	CDT'MRX'USED
6	# Ldevs cached	CDT'HUH'LDEVS
7	1st Cache device entry (entry number)	CDT'DISC'NERD
<b>X10</b>	# Words this DST	CDT'DST'WDRDS
<b>X1</b> 1	TRUE if stopcacke pending	CDT'STDP'PND
<b>X12</b>	# Sectors sequential fetch	CDT'SEQ'MI#FTCH
<b>X13</b>	# Sectors random fetch	CDT'RND'MI#FTCH
<b>X14</b>	TRUE if wait for physical post	CDT'FORCE'POST
X15	Nead of impeded queue (PIN)	CDT'STOP'QUEUE
<b>X16</b>		
	:	
7,27	•	

Disc Caching

COT'ENTRIES CDITENREES
The total number of CDT entries configured in this table (i.e. includes all three types of entries). The number of entries in the table will be:

1 entry for the header
+1 entry for each disc idev configured.
(CDT Device entries)
+1 entry for each DRQ configured.
(CDT Mapped Domain entries)

6.00.00 23- 6

This scheme insures that this table can never overflow (since an entry in the DRO table is always obtained before an entry in this table).

CDT'SIZE Size of each entry in the table.

CDT'FREE'COU#T
Total number of entries currently unassigned.

Table relative offset (i.e. Entry number \* entry size) of the first available entry.

CDT'FREE'TAIL
Table relative offset of the last available entry.

CDT'MAX'USED
The maximum number of entries in use at one time.

CDT'NUM'LDEVS
The number of ldevs currently cached.

CDT'DISC'HERD
The entry number of the first Device Entry.

CDT'DST'MDRDS
The total number of words in this data segment.

CDT'STDP'PND This value will be TRUE if there is a pending :STDPCAC#E.

CDT' SEQ' MINFTCH COI'SCY'NIMFICM
If there is a prefetch for a sequential read ("niss"), the size of the prefetch is delinited by the extent size of the file. Within this limitation, the prefetch is equal to the greater of two sizes:

1) Requested size.

2) The largest integer multiple of the request size that is smaller than the value found in this cell.

The default value is 96 sectors. (This value may be changed  $\ensuremath{\text{via}}$  :CRCHECONTROL).

CDT'RND'MINFTCH This is the same as LUI SEQ NIMP[LH except that it's for random access. The default value is 16 sectors. (This value may be changed via :CRCHECONTROL).

### Diec Caching

COT'FORCE POST

then this value is TRUE, all writes will 'block" until the physical update on disc completee. The system default is FRUSE. (Can be altered via :CRCHECOMTROL).

CDT'STOP QUEUE

If COT'STOP PENOING is TRUE this will be the PIN number of the head pin of
the processes impeded until the :STOPCACHE completes.

Disc Saching

### Device Entry

0	Next idev entry (entry number)	COTINE IN GEN
1	Prev ldev entry (entry number)	COTTOETPREVILOEV
2		COTIDETLOEV
3		COT DE'MARG PAGES
4		TAT 1658H13C11CO
5	Head of Happed domain (entry number)	TOTIOE: KAPOIHEAD
8	Tail of mapped domain (entry number)	1181'0981'3I TG3
7	# Disc domain regions for this device	CDT'DE'REGIOMS
210	Temory address of head	CDT'DE'REG'HD
	tached disc domain	
112	Memory address of tail	COT'DE'REG'TL
	cached disc domain	
<b>114</b>	– W Read hits –	CDT'DE'RHII
X16	- # Write bits -	CDT19E1WHIT
Z20	- # Read n_sses -	adtideiRhISS
<b>222</b>	- # Write Hisses -	2271361#4182
724	- # Stops -	2071/36/1768
7.26	Temory address of Last	DDT SE SCHWPT
	referenced potain	

Disc Caching

COT'DE'NEXT'LDEV
The entry number of the next Device Entry.

COT' DE' PREV' LDEV

The entry number of the previous Device Entry.

CDI'DE'LDEV
The Ldev number for this cached device.

COT' DE' MAPD' PAGES

Total number of main memory pages allocated to disc domains for this cached device. This includes napped and unmapped regions. (1 main memory page = 128 words).

The total number of Mapped Domain entries associated with this Device Entry.

COT'DE'MAPO'HERO

The entry number of the first Mapped Domain entry for this device.

The entry number of the last Mapped Domain entry for this device.

CDT'DE'REGIONS

The total number of disc domain regions for this ldev (includes mapped and unmapped regions).

CDT'DE'REG'HO

COTTOE REGIME

Theory address to the head region of the disc domain linked list. Disc domain regions are linked in order based on the disc address they represent (i.e. small disc address at head, large disc address at tail). This address will not point to the region base (RB), but to the next domain (MD) field of the region header. (This is to facilitate the use of the LLSH instruction).

COT'OF'REG'TL

Theory address of the tail region of the disc domain linked list. This address will be of the previous domain (PD) field of the region header.

Total number of times that a read was requested and the requested disc domain was present in memory - i.e. a read "hit". This means that the read completed uithout performing any I/O (to fetch the domain). Thus this is actually the number of read I/Os eliminated. This value will reset to zero on overflow.

CDITOCHMII Total number of times that a write was requested and the requested disc domain was present in memory — i.e. a write "hit". If there was no other write pending to the "hit" domain, then the process would continue as soon as the cache move completes — thus eliminating a block for I/D. Otherwise, the process would block waiting for the first write to complete. This value will reset to zero on overflow.

### Disc Caching

CONTEXTMISS Total number of times that a read was requested and the requested dist ionain was not in memory — i.e. a read "miss". This means that the requested disc onain had to be fetched into memory before the read could complete — thus potentially blocking the process. This value will reset to zero on overflow.

CDITOETHMISS
Total number of times that a white was requested and the requested disc domain was not in remony inle, a white muss. This does not mean that the process would block until the disc domain is fetched as is the case for neads. Rather, a free memory region would be obtained to be the destination of the nache hove. This disc domain would then be posted in the background (unless overmidden wis "FERGEOWROL or FEETHODE) allowing the process to continue without blocking. This value will reset to zero on overflow.

CDT'DE'STOP
Total number of times that a process had to block on a cache transfer. Will reset to zero on overflow.

COT' DE'SCRNPT

COPPORTSCRAP!
The memory address of the last region looked at on a scarch. This address will be of the mext domain (ND) field of the region beasen. This value will be used along with IOTIDETRETHD to determine where to start the reak search for a cached disc domain Rt times it will be more efficient to start with this address since the disc domain requested may be of a nighter disc address than found in this region header, rather than always starting the search with CDTIDETRECTHD.

### Mapped Domain Entry

۰į	Prev mapped domain entry (entry number)	CDT' NO' PREV
1	Next mapped domain entry (entry number)	CDT, HO, HEXT
2	Start sector	COT'MO'SECTOR
¦	aodress	
4	Last eector	CDT'MD'END'SECTOR
	- addrese	
6	R	CDT'MD'FLAGS
7	W Reads pending	CDT'HO'READ'CHT
210	N Writee pending	COT'HO'HRITE'CHT
X11	fock marting	CD1,WD.FKD,CD1
X12		CO1.UD.IUBED.HD
<b>X13</b>	Head of active LDR	CDT'MD'LDR'HERD
X14		COT'HO'HEH'ROR
	if present	
X16	DRQ for this napped domain	COT'HO'DISCREG
<b>X17</b>	W Flushing CDTs	COT'MO'LK'CHT
X20	Ldev for thie mapped domain	COT.NO.FDEA
X21	Head impeded queue (PIN)	COT'NO'IMPEDED
X22	Device entry (entry number)	COT. NO. DE
X23		
	:	}
127	!	   

G.00.00 23- 13

Disc Caching

CDT'NO'PREV Entry number of the orevious mapped domain entry for this device.

Entry number of the next mapped domain entry for thie device.

CDT'mD'SECTOR
The etarting disc sector address representing this mapped domain entry.

COT'MO'END'SECTOR The ending disc sector address representing this mapped domain entry.

COTINDIFERGS flags describing the state of this mapped domain entry and the region as-sociated with it:

- (0:1) Rheent.
  Region is not present in memory.
  (1:1) Ini.
  Region is already In-Motion-In. (Set when the fatch for this cached region is initiated).
- this cached region is initiated.

  (2:1) IND.
  Region is In-Motion-Dut. (Set by STARTOBJURITE when performing the background poet of a cached region).

  (3:1) INSS.
  This died domain wae not precent and had to be prefetched.

  (4:1) LOCK. Not used.

  (5:1) Fulf.
  Forced Write In Progress. Region wae forced out of memory to make room for another object.

  (6:1) RNC

- to make room for another object.

  Recover Overlay Candidate. Region may be forced out of
  memory to make room for another object. However, if this
  region is referenced again it can be recovered.

  VIRGIN.

  Llean region in the write state. Cleared as soon as a now (6:1) -
- (7:1) -
- VIRGIN.
  Clean region in the write state. Cleared as soon as a nove complete. (I.e. if this but is on, then a write can complete inmediately. Otherwise the write will have to wait until the current write completes the physical post).

  HOPOSI.

  Set when the CDT is being posted out as a result of a write request that did not want to wait for the physical poet to complete. This will be cleared by the cache completor when the physical post completes. (The le used to insure that a cache move for any subsequent write request will not be serviced until the physical poet completes.)

  552.
- SET. Set if doing sequential I/D. When the request for the last area of this disc domain is complete, this domain will be made a ROC. Not used.

(10:3) - Not used. (13:3) - STRIE 0 - RVRIL. CDT is an available entry.

G.00.00

Disc Caching

- 1 RERO. Only read LDR(s) are attached.
  2 WRITE. Write LDR(s) and poecibly read LDR(s) are attached.
  3 FLUSM. CDT ie being flushed out.
  4 LDCK. Unused.

CDT'MD'READ'ENT
The number of LDRs attached that are for reads (move not complete).

COT'HO'URITE'CHT The number of LDRs attached that are for writes. NOTE: This count will not be decremented until both the cache nowe end the physical write completes. Note: the cache nove completes, the LDR will be dequeued from the CDT.

COL. MO. IND. COL

COT'MO'INPED'HD
The first LDR that is impeded. (I.e. the CDT is in a write state already and another write is attached. The econd write will be placed in this queue until the first write complete.)

CDT'MO'LDR'HERD
The first LDR that is on the active liet for this CDT.

COT'NO'NEN' ADDR

The nemory address (region base) for this napped diec domain, if present.

COT'MO'DISCRED

COLTENTIAL TABLE 1 INDEX ASSOCIATED WITH THE MADE TO THE MILE THE DESCRIPTION OF THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE MILE THE

CDT'MO'LK'CHT Not used.

COT'NO'LDEV

The ldev number for this napped donain.

CDT'MO'IMPEDED
The PIM for the first procees impeded on this napped disc domain. Processes get impeded here when they do URIFORID when their LDR is on the CDT impeded queue and the Mapped Domain is currently being written out. (This will also happen upon a 'SIDPCRIME to force all LDRs to complete.) As soon as the physical post of the Mapped Domain is complete, all processes impeded here will be awakened.

The entry number for the Device entry that this Mapped Domain entry is associated with.

Diec Caching

### Logical Disc Request Table

%1017 Pointer to Logical Disc Request Table

NOTE: This table is really part of the ORQ (Chapter 13). Any entry with the logical request bit set in the flags will conform to this for-mat and not the format of the etandard DRQ.

Logical disc requests entriee are used to manage requeste detueen the requesting process and a mapped disc domain. They are the counterpart of disc requeste entries used to manage physical I/O requests between a process and a disc. These entriee are kept as part of the DRQ Table, but uill never be queued to the disc's DIT, inetead they will be queued to the mapped disc domain CDT entry. LDR entriee may only be placed onto the following queues:

- The CDT active list.
   The CDT impeded LDR list.
   The Disabled Disc Request. (This will only happen if the buffer segment is absent when the logical I/D (cache move) is attempted.)

: LDRs are eingly linked onto the CDT queuee and doubly linked onto the disabled disc request queue.

### Logical Disc Request Entry

	3 4 5 6 7 8 9 01 2 3 4 5	
-0	//  SI II 8  DI DI SI CI M /  CI DI LI II /// BB IO LI OI OI EL DI OI/I UI II DI NI /// IF AI CI EL PI II II/I AI LI /// IF AI CI EL PI II II/I AI LI /// IF KK II OI AI QID/I AB SI AI OI /// IF EL EL SI LI UI OI/I EL LI EL CI /// IF II II II EL II/I QI EL II// /// II II II II EL II/I QI EL II//	LDR'FLAGS
1	NDDA of extent linit	LDR'L'HODA
2	Ldev	LDR' LDEV
3	Mapped Donain CDT entry number .	LDR'CDT
4	S  DST number	LDR'BUFDST
5	Offset into DST	LDR' BUFADR
6	Strategy   Function	LDR'STRAT'FUNC
7	Count/Xlog/Control returns	LOR' COUNT
<b>210</b>	P1	LDR'PRRM1
<b>%11</b>	P2	LDR'PARM2
<b>%12</b>	Qualifier   Status	LOR'STRTQ
<b>%</b> 13	PIN number	LDR'PCB
214	Prev. LDR in queue (table relative)	LDR'PREVQ
<b>%15</b>	Next LDR in queue (table relative)	LDR'NEXTQ
<b>216</b>	HODA of extent base	LDR'B'HODR
<b>X17</b>	LODA of extent base	LDR'B'LODA
X20	LODA of extent limit	LDR'L'LODA

G.00.00 23- 17

LOR'FLAGS

LORTHUMS
Flags.

(0:3) - Not used.

(3:1) - SBUF.
Set if request is to/from a System Buffer.

(4:1) - IDHRKE.
Set if system should make up the process when the logical 1/0 completes.

- BLUCKED. I/O completes.

(5:1) - BLOCKED.
Set if the process wants to wait for the logical disc request to complete.

(6:1) - DONE.
Set when the logical disc request is complete and the process will be awakened (if IOWAKE is set)

(7:1) - DO'POST.
Set if the caller wants to be waited until the physical post to disc completes. Only valid for write requests.

(8:1) - SERIGH'POST.
Set when the physical post should be through the Global Serial Write queue.

Serial Write queue.

Serial Mrite queue.

(9:1) - CDT'QUEUED.

This request has been queued - either onto the CDT active queue (see CDT happed Domain entries) or onto the disabled disc request list.

(10:1) - MOVE'DOME.

The move has been completed, but the process won't be awakened until the DONE bit is set.

(11:1) - Not used.

(12:1) - CUMPREQ.

Set if this request is the current/active request.

13:1) - DISRBIE.

(13:1) -  $\frac{\text{Set II}}{\text{DISRBLE}}$ . Set if the request is disabled.

(14:1) - LORPREG.
Set if thus is a logical disc request.
(15:1) - LORTINLOC.
Set if Mapped Domain CDT entry is in process's locality list.

LDR'L'NDDR
The Nigh Order Disc Address of the extent limit. (See note with LDR'B'HODR).

LDR'LDEV
The ldev for this request.

LDR'CDT The CDT number for the Mapped Domain entry associated with this request.

Data Segment number for the target of the logical I/O request. If bit zero is set, then this is the process's stack.

LDA'BUFFOR Offset within the DST (above) for the target address. If the DST is the process's stack, then this address will be DB relative.

G.00.00 23- 18

Disc Caching

LDR'STRAT'FUNC

LDR'STRAT'FUNC
(0:8) - Strategy
0 - Unknown caller
1 - Unknown File System
2 - Spooler
3 - Directory
4-7- Unknown caller
8 - Gennessage
9 - File Systen, Duiesce I/O
10 - File Systen, Dequential, No Buf
11 - File Systen, Direct, No Buf
12 - File Systen, Direct, Buffered
13 - File Systen, Direct, Buffered
14 - File Systen, Direct, Buffered
14 - File Systen, SRM
15 - File Systen, IMRGE

(8:8) - Function

0 - Read 1 - Write

On initiation, this specifies the requested transfer count (+words, -bytes). Rt completion of the request, this contains the actual transmission count (+words, -bytes).

LDR'PRRM1
This is the Nigh Order Disc Address of the requested disc sector.

LDR'PRRM2
This is the Low Order Disc Address of the requested disc sector.

LDR'STRTQ Uniform status returns.

LDR'PCB PIN of the requesting process.

LDR' PREVQ

Table relative index of the previous LDR in the queue. (NOTE: LDRs are singly linked on the CDT queues, and doubly linked on the disabled disc request queue).

LDR'NEXTQ Table relative index of the next LDR in the queue.

DON'D HUBH The Nigh Brder Disc Address of the extent base. (Used when the logical disc request is through the file system. Caching uses this information when searching memory for a "hit" on a cached domain).

 $\mbox{LDR}^{\prime}\mbox{B}^{\prime}\mbox{LDD}\mbox{R}$  The Low Order Disc Rddress of the extent base. (See note above).

The Low Order Disc Address of the extent limit. (See note above).

# READER COMMENT SHEET

# MPE V Tables Manual for MPE V/E, Version G.00.00

32033-90010

September 1984

We welcome your evaluation of this manual. It is one of several that serve as a reference source for HP 3000 Computer Systems. Your comments and suggestions help us to improve our publications and will be reviewed by appropriate technical personnel. HP may make any use of the submitted suggestions and comments without obligation.

Is this manual technically accurate?	Yes [] No []	(If no, explain under Comments, below.)		
Are the concepts and wording easy to understand?	Yes [] No []	(If no, explain under Comments, below.)		
Is the format of this manual convenient in size, arrangement and readability?	Yes [] No []	(If no, explain or suggest improvements under Comments, below.)		
Comments:				
We appreciate your comments and suggestions. This form requires no postage stamp if mailed in the U.S. For locations outside the U.S., your local HP representative will ensure that your comments are forwarded.				
	Date:			
77.014				
FROM:				
Name				
Company				
Address				

FOLD

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

# **BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 1070 CUPERTINO, CALIFORNIA

POSTAGE WILL BE PAID BY ADDRESSEE

Documentation Manager/47U-91 Hewlett-Packard Company Computer Systems Division 19447 Pruneridge Avenue Cupertino, California 95014

FOLD

FOLD